FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

Problemas de regresión e clasificación usando SVM

Ángela Dono Caramés

2018/2019

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

GRAO DE MATEMÁTICAS

Traballo Fin de Grao

Problemas de regresión e clasificación usando SVM

Ángela Dono Caramés

Xullo 2019

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

Traballo proposto

Área de Coñecemento: Estatística e Investigación Operativa
Título: Problemas de regresión e clasificación usando SVM
Breve descrición do contido
Neste traballo revisaranse as principais aplicacións das Support Vector Machines (SVM) no eido da regresión e a clasificación. Este procedemento que pode entenderse como un problema extendido de programación lineal é unha das ferramentas máis populares en Data Mining. Unha introducción pode atoparse nas referencias seguintes: Abe, S. (2010) Support Vector Machines for Pattern Classification. 2ed. Springer. Steinwart, I. y Christmann, A. (2008). Support Vector Machines. Springer
Recomendacións
Soltura coa linguaxe de programación R
Outras observacións

Índice xeral

Resumo

O noso obxectivo é estudar un tipo de algoritmos de aprendizaxe coñecidos como support vector machines. Comezamos por introducir un caso sinxelo de clasificación binaria, o clasificador de marxe máxima, que empregamos para clasificar datos linealmente. Despois presentamos o concepto de función kernel, importante á hora de realizar clasificacións non lineais e introducimos as variables slack para mellorar a capacidade de predición. A continuación, estudamos dous métodos que nos permiten traballar con problemas de clasificación multiclase.

No segundo capítulo xeneralizamos as support vector machines aos problemas de regresión mediante o uso de funcións de perda ε-insensitivas, que se basean en ignorar os erros se estes son o suficientemente pequenos.

Ao final do traballo ilústrase a implementación en R destes algoritmos con dous exemplos reais.

Abstract

Our objective will be studying a type of leaning algorithms known as support vector machines. We start by introducing a simple example of binary classification, the maximal margin classifier, which we use to separate data linearly. Then we present the kernel functions, which become really important at classifying data in a non-linear way, and slack variables are introduced to improve the prediction ability. Next, we study two methods that allow us to apply the algorithm to non-binary classification.

In the second chapter, the concept of support vector machine is generalized to apply it to regression problems by defining an ε-insensitive loss function that ignores errors smaller than a certain constant.

Finally, we illustrate how to apply these algorithms in R by using two real examples.

Introdución

Nos últimos anos, atopámonos cada vez con máis frecuencia problemas ou tarefas que non podemos resolver cun ordenador programándoo da forma habitual. Isto débese a que non somos quen de expresar de forma precisa como obter a saída desexada en función da entrada que se proporciona. Isto ocorre nunha ampla variedade de situacións, dende tratar de modelar reaccións químicas complexas nas que non se coñece de xeito preciso a forma en que os reactivos interactúan entre sí ata o recoñecemento de imaxes ou de voz de forma automática. Nestes casos, o que se adoita facer é intentar que o propio ordenador aprenda a dependencia entre as entradas e as saídas proporcionándolle unha cantidade suficiente de exemplos, do mesmo xeito en que a un neno se lle ensina a recoñecer certos obxectos dicíndolle cal é o nome de cada cousa en lugar de darlle unha lista de características que definen o obxecto en cuestión. Esta forma de afrontar o problema coñécese como *metodoloxía de aprendizaxe*. O habitual será que teñamos un conxunto de datos e queiramos atopar certa estrutura neles que nos permita clasificalos segundo as súas características ou vendo a relación que presentan uns con outros. Segundo a cantidade de información que teñamos a cerca deste conxunto de exemplos respecto á solución que queremos obter, podemos clasificar a aprendizaxe en dous tipos:

- A *Aprendizaxe supervisada* é aquela na que temos un conxunto de exemplos para os cales coñecemos toda a información, incluída a variable na cal estamos interesados, e dada unha nova observación para a cal non coñecemos o valor desta variable queremos determinalo comparando o resto das súas características cos demais exemplos. Esta situación preséntase, por exemplo, se temos un conxunto de datos de individuos dos que coñecemos certas variables como a súa idade, sexo, presenza ou non de certos síntomas, etc. e se padecen unha certa enfermidade. Dado un novo individuo do que coñezamos os valores destas variables queremos obter a probabilidade de que padeza dita enfermidade.

- Na *Aprendizaxe non supervisada*, a diferenza da anterior, non coñecemos o valor da variable desexada para ningunha das observacións que temos. Isto ocorre cando quere-

mos dividir o conxunto de exemplos agrupando aqueles que presenten características similares. Neste caso temos liberdade para determinar o número de grupos que imos empregar e que variables terán máis peso á hora da clasificación. Un exemplo práctico da aprendizaxe non supervisada é a detección de datos anómalos.

Neste traballo consideraremos só dous problemas dentro da aprendizaxe supervisada: a regresión e a clasificación. A *regresión* é un proceso estatístico que busca atopar a dependencia dunha variable Y, que chamamos *variable resposta* con respecto a outra, X, que poderá ter unha ou máis compoñentes independentes e que coñecemos co nome de *variable explicativa*. O obxectivo é construír un modelo que reflicta como varía Y cando cambia o valor dunha das compoñentes de X e o resto das compoñentes se manteñen constantes. Unha vez construído o modelo podemos empregalo para realizar predicións pois dada unha nova observación da variable explicativa X podemos estimar o valor de Y que lle correspondería.

A regresión aparece por primeira vez a principios do século XIX con traballos de Legendre, e posteriormente Gauss, no método de mínimos cadrados. Aínda así, o termo "métodos de regresión" non foi acuñado ata finais do século por Galton no seu estudio sobre a dependencia da altura dos fillos con respecto á dos seus pais, onde observou que aínda que os descendentes de persoas de grande estatura eran polo xeral tamén altos, non o eran tanto coma os seus pais senón que tendían a regresar cara a media da poboación.

A *clasificación* podémola entender como un caso particular da regresión na cal a variable resposta Y toma só un número finito de valores. Así, cada valor x da variable X terá asociado un valor y ao cal chamaremos *etiqueta*, co que podemos dividir as observacións en clases dependendo de cal sexa a súa etiqueta. Co modelo de regresión, que se dirá agora de clasificación, podemos predicir a clase á que pertencerá unha nova observación x_0. Na práctica este problema aparece en moitas situacións. Por exemplo, podémolo atopar á hora de clasificar o correo basura, pois estaríamos a separar o correo electrónico en dúas clases. Outro caso no que nos podemos atopar a clasificación sería tratar de determinar a que partido político votará unha determinada persoa coñecida certa información sobre ela.

Tanto no caso da clasificación coma no caso máis xeral da regresión, asumiremos que existe unha función que regula a saída de datos a partir dos datos de entrada, que chamamos *función obxectivo* de forma xeral e *función de decisión* no caso particular da clasificación. O noso obxectivo será, por tanto, atopar algoritmos de aprendizaxe cos cales estimar esta función entre un conxunto de funcións preestablecidas que determinamos de antemán e coñecemos como *hipóteses*. Porén, non buscamos tan só que a hipótese que escollemos represente ben os exemplos de adestramento, senón que buscamos ademais que sexa quen de clasificar correctamente datos que non pertenzan ao seu conxunto de observacións orixinais.

Isto coñécese como capacidade de *xeneralización* é será o que procuremos optimizar. Para acadar este obxectivo, teremos que reducir na medida do posible o espazo de hipóteses, pois se permitimos que estas teñan unha complexidade moi elevada pode suceder que a función se adapte en exceso aos exemplos de adestramento sen que clasifique correctamente datos novos, que é o que coñecemos como *sobreaxuste*. Isto pode ocorrer, por exemplo, se dado un conxunto de observacións que queremos clasificar lle asignamos a cada unha delas unha etiqueta distinta. Deste xeito teríamos unha división nos datos que non nos axudaría a clasificar novas observacións.

Neste traballo, centrarémonos nun tipo de algoritmos de aprendizaxe coñecidos como *máquinas de vectores de soporte* ou *support vector machines* (SVM). As SVM aparecen por primeira vez a finais dos 70, co traballo de Vladimir Vapnik, e popularízase na década dos 90 grazas aos bos resultados que acadan nunha gran variedade de problemas. Este tipo de algoritmo aparece orixinalmente como un algoritmo para a clasificación, que posteriormente se modifica para podelo aplicar tamén á regresión. As SVM que se empregan na clasificación baséanse principalmente en transformar os exemplos a un espazo de alta dimensión (posiblemente infinita) e despois clasificalos neste novo espazo de forma lineal, é dicir, separando o espazo en dous mediante un hiperplano, o cal pode resultar en clasificacións non lineais no espazo de partida. Ademais, para facer esta clasificación, escóllese como hiperplano separador aquel que deixe máis marxe entre os datos. Isto parte da idea de que, se ben todos os hiperplanos que conseguen separar os datos os axustan igual de ben (de feito axústanos de forma perfecta), ao considerar novos exemplos a súa capacidade de xeneralización pode variar, e un hiperplano que deixe unha gran marxe adoita realizar boas predicións. Estas ideas explicaranse con máis detalle nos capítulos seguintes, nos cales veremos como se foi modificando este tipo de algoritmos, dende os casos máis sinxelos que se introduciron ata as SVM empregadas na actualidade, para resolver algunhas das carencias que tiñan os modelos máis primitivos e que dificultaban a súa implantación na práctica.

Capítulo 1

SVM para a clasificación

1.1. Clasificación binaria

1.1.1. Caso linealmente separable

Comenzaremos por introducir o algoritmo máis sinxelo dentro das SVM, o coñecido como *clasificador de marxe máxima* ou *hard-margin support vector machine* na terminoloxía inglesa. Como veremos deseguido, este algoritmo emprega unhas hipóteses moi restritivas sobre o conxunto de datos, o cal imposibilita case por completo o seu uso na práctica ao non termos na vida real exemplos tan simples. Non obstante, é o primeiro algoritmo en ser introducido e ten gran importancia pois veremos máis adiante que outros SVM máis complexos están baseados nel.

Supoñamos que temos un conxunto de l observacións nun espazo n-dimensional X, ao que chamaremos *espazo de entrada*,

$$\mathbf{x}_1 = \begin{pmatrix} x_{11} \\ \vdots \\ x_{1n} \end{pmatrix}, \dots, \ \mathbf{x}_l = \begin{pmatrix} x_{l1} \\ \vdots \\ x_{ln} \end{pmatrix}$$

e estas observacións pódense clasificar en dous tipos, que denotaremos 1 e -1 respectivamente, é dicir, que a cada $\mathbf{x}_i$ lle corresponde unha *clase* ou *etiqueta* $y_i \in Y = \{-1, 1\}$. Denotaremos por $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)) \subset (X \times Y)^l$ o conxunto destas observacións que coñecemos co nome de *conxunto de datos de adestramento*. Consideramos ademais unha observación a maiores, $\mathbf{x}_0 = (x_{01} \ \dots \ x_{0n})^T$. O noso obxectivo é obter unha función de decisión baseada nos exemplos de adestramento S que clasifique correctamente esta nova observación. En primeiro lugar, supoñemos que os exemplos son *linealmente separables*, ou sexa, que existe un hiperplano capaz de separar perfectamente os exemplos de S, deixando a un lado aqueles que teñen unha etiqueta positiva, 1, e ao outro os de etiqueta negativa,-1.

Neste caso existirán en xeral infinitos hiperplanos nestas condicións xa que, dado un hiperplano separador, normalmente podemos facer pequenas translacións ou rotacións sen que o hiperplano entre en contacto con ningún exemplo e por tanto este novo hiperplano resultante segue sendo separador. Na Figura 1.1 podemos ver varios destes hiperplanos no caso 2-dimensional.

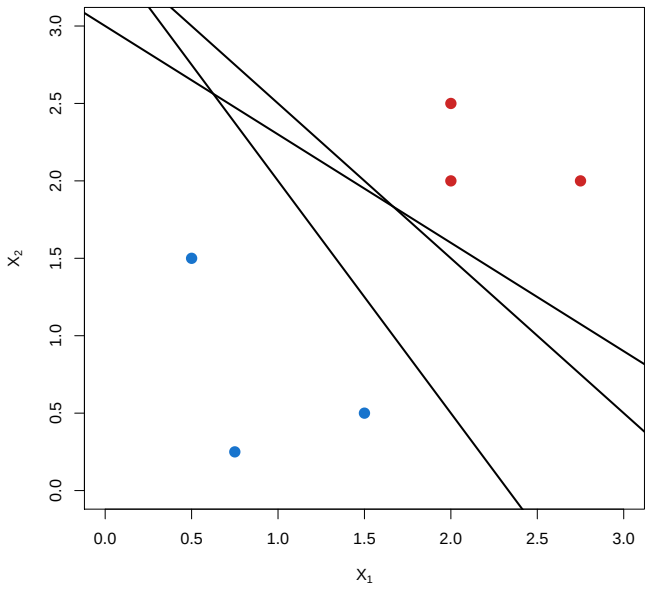


Figura 1.1: Un conxunto de datos con etiquetas 1 e -1, que representamos por puntos vermellos e azuis respectivamente, pódense separar dependendo da clase á que pertenzan mediante moitos hiperplanos. Móstranse tres, dos infinitos que serían posibles, que están representados por unha liña negra.

Sexa H un destes hiperplanos separadores. Así, para clasificar unha nova observación $\mathbf{x}_0$ bastará determinar se H deixa a $\mathbf{x}_0$ do lado dos exemplos positivos ou dos negativos (se coincide que $\mathbf{x}_0$ está contido en H, entón consideraremos que o dato non pode ser clasificado polo hiperplano e diremos que x_0 é *non clasificable* por H). Para isto consideramos un vector $\mathbf{w} = (w_1 \ \ldots \ w_n)^T$ perpendicular a H, entón $\mathbf{x}_0$ será positivo se $\langle \mathbf{w}, \mathbf{x}_0 \rangle \geq c$ para certo $c \in \mathbb{R}$ e negativo noutro caso. Pero esta regra de decisión é equivalente a

$$\langle \mathbf{w}, \mathbf{x}_0 \rangle + b \geq 0, \tag{1.1}$$

pois basta tomar $b = -c$. Temos por tanto que a función de decisión para clasificar o novo dato vén dada polo signo da función $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x}_0 \rangle + b$. Non obstante, ademais de

fixarnos en se $f(\mathbf{x})$ é positivo ou negativo, podemos ter en conta a súa magnitude para determinar como de preto está do hiperplano. Isto permítenos dalgún xeito cuantificar a confianza que temos na clasificación que realizamos, pois podemos estar máis seguros de que clasificamos correctamente un dato que se atope lonxe do hiperplano mentres que a clasificación dun dato que se atope moi preto do hiperplano pode resultar máis dubidosa, xa que algunha leve alteración das súas compoñentes podería facer que este dato pasase ao outro lado do hiperplano. De feito, o erro que presenta un clasificador lineal á hora de clasificar datos novos pode ser acotado en función da marxe que deixa o hiperplano, é dicir, da menor distancia entre este e os exemplos de adestramento.[1] Por esta razón, é lóxico que nos centremos no *hiperplano de marxe máxima*, tamén coñecido como *hiperplano separador óptimo*, pois é o que minimiza esta cota de erro.

Na Figura 1.2 aparecen representadas as marxes correspondentes ao hiperplano óptimo. Vexamos a continuación como podemos determinar este hiperplano.

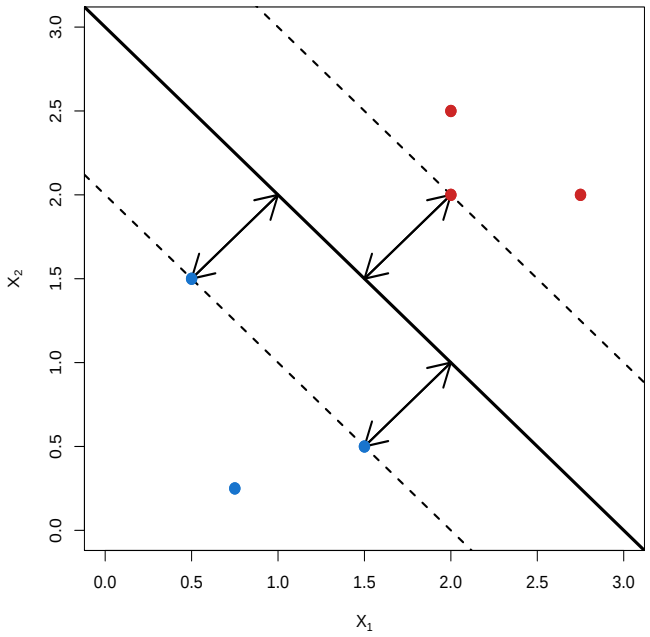


Figura 1.2: O hiperplano separador é o que deixa maior marxe, que é a menor distancia entre o hiperplano e os exemplos. Representamos esta distancia mediante frechas nos puntos máis próximos ao hiperplano e a marxe por liñas descontinuas a ambos lados do hiperplano.

[1] Unha formalización desta cota pódese atopar no cuarto capítulo de Cristianini, N. and Shawe-Taylor, J. (2000) [2], páx.63

Como tiñamos por hipótese que os exemplos de adestramento son linealmente separables, temos que a súa distancia ao hiperplano é positiva, logo ningún dato se atopa no propio hiperplano, co cal

$$\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle + b > 0, & \text{se } y_i = 1 \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b < 0, & \text{se } y_i = -1. \end{cases}$$

Agora ben, ao termos un número finito de datos, podemos tomar $\beta > 0$ como a mínima distancia entre os exemplos e H, e por tanto

$$\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geq \beta, & \text{se } y_i = 1 \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leq -\beta, & \text{se } y_i = -1. \end{cases}$$

Se dividimos ambas expresións entre β e renomeando $\mathbf{w}/\beta$ por $\mathbf{w}$ e b/β por b, temos

$$\begin{cases} \langle \mathbf{w}, \mathbf{x}_i \rangle + b \geq 1, & \text{se } y_i = 1 \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b \leq -1, & \text{se } y_i = -1. \end{cases}$$

Ademais, podemos simplificar e escribir isto mediante unha soa ecuación pois o sistema anterior equivale a

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 \geq 0 \quad \forall i = 1, \dots, l. \tag{1.2}$$

Nótese que para aqueles exemplos que están a menor distancia de H dáse a igualdade, mentres que para todos os demais tense a desigualdade estrita.

Para calcular a marxe do hiperplano podemos escoller dous exemplos $\mathbf{x}_1$ e $\mathbf{x}_2$ con etiquetas -1 e 1 respectivamente e que se atopen a distancia mínima do hiperplano, é dicir, que verifiquen $\langle \mathbf{w}, \mathbf{x}_1 \rangle + b = -1$ e $\langle \mathbf{w}, \mathbf{x}_2 \rangle + b = 1$. Entón, podemos considerar un vector unitario perpendicular a H, como por exemplo $\mathbf{w}/\|\mathbf{w}\|$, e tense que a proxección de $\mathbf{x}_2 - \mathbf{x}_1$ sobre este vector é precisamente dúas veces a marxe do hiperplano. Isto pódese apreciar na Figura 1.3. Por tanto, aplicando (1.2), tense que a expresión da marxe vén dada por:

$$\begin{aligned} \gamma &= \frac{1}{2}\left\langle \mathbf{x}_2 - \mathbf{x}_1, \frac{\mathbf{w}}{\|\mathbf{w}\|} \right\rangle = \frac{1}{2}\left(\left\langle \mathbf{x}_2, \frac{\mathbf{w}}{\|\mathbf{w}\|} \right\rangle - \left\langle \mathbf{x}_1, \frac{\mathbf{w}}{\|\mathbf{w}\|} \right\rangle\right) \\ &= \frac{1}{2\|\mathbf{w}\|}\left(\langle \mathbf{w}, \mathbf{x}_2 \rangle - \langle \mathbf{w}, \mathbf{x}_1 \rangle\right) = \frac{1}{2\|\mathbf{w}\|}\left(1 - b - (-1 - b)\right)) = \frac{1}{\|\mathbf{w}\|}. \end{aligned} \tag{1.3}$$

En consecuencia, para atopar o hiperplano de marxe máxima temos que maximizar $1/\|\mathbf{w}\|$, o cal é claramente equivalente a minimizar $\|\mathbf{w}\|$, que a súa vez é o mesmo que

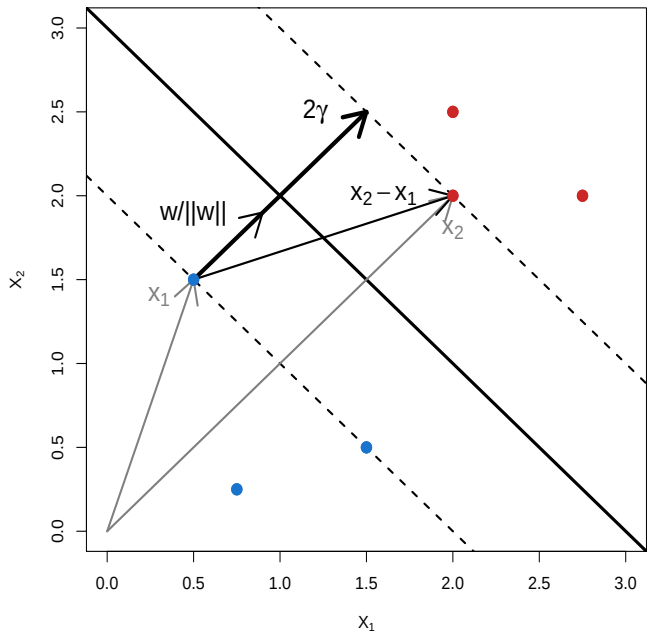


Figura 1.3: En gris aparecen representados os vectores de posición de dous exemplos $\mathbf{x}_1$ e $\mathbf{x}_2$ con etiquetas distintas e que están a distancia mínima do hiperplano. O vector de cor negra que os une é a súa resta, $\mathbf{x}_2 - \mathbf{x}_1$. Para calcular a marxe do hiperplano, que denotamos por γ, bástanos con facer o produto interior deste último vector $\mathbf{x}_2 - \mathbf{x}_1$ cun vector unitario perpendicular ao hiperplano, $\mathbf{w}/\|\mathbf{w}\|$. Con isto obtemos a proxección de $\mathbf{x}_2 - \mathbf{x}_1$ sobre a dirección de $\mathbf{w}$, que se corresponde con dúas veces a marxe do hiperplano e está representado na figura por un vector máis resaltado ca o resto.

minimizar $\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle$. Noutros termos, bastaría con resolver o seguinte problema de optimización:

$$\begin{aligned} &\text{minimizar} && \langle \mathbf{w}, \mathbf{w} \rangle \\ &\text{suxeito a} && y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 \geq 0, \quad \forall i = 1, \dots, l. \end{aligned} \tag{1.4}$$

Veremos agora como se constrúe o problema dual, que será moito máis sinxelo de resolver e por tanto máis eficaz na práctica, e que ademais nos aporta determinada información. Para isto, necesitaremos certas nocións de dualidade que recordamos nos seguintes resultados, dos cales omitiremos a demostración.[2]

[2] O caso lineal destes resultados xa está probado na materia de Programación Lineal.

Definición 1.1. Dado un problema de optimización con dominio $\Omega \subset \mathbb{R}^n$,

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{w}), && \mathbf{w} \in \Omega \\ \text{suxeito a} \quad & g_i(\mathbf{w}) \leq 0, && \forall i = 1, \dots, k, \\ & h_i(\mathbf{w}) = 0, && \forall i = 1, \dots, m, \end{aligned}$$

definimos a súa *función lagranxiana xeneralizada* como

$$L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\mathbf{w}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{w}) + \sum_{i=1}^{m} \beta_i h_i(\mathbf{w}) = f(\mathbf{w}) + \boldsymbol{\alpha}^t g(\mathbf{w}) + \boldsymbol{\beta}^t h(\mathbf{w}).$$

onde as variables $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$ coñécense como *multiplicadores de Lagrange*.

Ademais, definimos o seu *problema lagranxiano dual* como:

$$\begin{aligned} \text{maximizar} \quad & W(\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ \text{suxeito a} \quad & \alpha_i \geq 0, \quad \forall i = 1, \dots, l, \end{aligned}$$

sendo $W(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \text{ínf}_{\mathbf{w} \in \Omega} L(\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta})$.

Teorema 1.2 (Kuhn-Tucker)**.** *Dado un problema de optimización con dominio convexo* $\Omega \subset \mathbb{R}^n$*,*

$$\begin{aligned} \text{minimizar} \quad & f(\mathbf{w}), && \mathbf{w} \in \Omega \\ \text{suxeito a} \quad & g_i(\mathbf{w}) \leq 0, && \forall i \in 1, \dots, k, \\ & h_i(\mathbf{w}) = 0, && \forall i \in 1, \dots, m, \end{aligned}$$

con $f \in C^1$ *convexa e* g_i, h_j *aplicacións afíns* $\forall i = 1, \dots, k$, $\forall j = 1, \dots, m$*. Entón, para que* $\hat{\mathbf{w}}$ *sexa un óptimo, é necesaria e suficiente a existencia de* $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ *tales que:*

$$\begin{aligned} \frac{\partial L(\hat{\mathbf{w}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})}{\partial \hat{\mathbf{w}}} &= 0 \\ \frac{\partial L(\hat{\mathbf{w}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})}{\partial \beta} &= 0 \\ \hat{\alpha}_i g_i(\hat{\mathbf{w}}) &= 0, \quad i = 1, \dots, k \\ g_i(\hat{\mathbf{w}}) &\leq 0, \quad i = 1, \dots, k \\ \hat{\alpha}_i &\geq 0, \quad i = 1, \dots, k. \end{aligned}$$

A terceira condición, $\hat{\alpha}_i g_i(\hat{\mathbf{w}}) = 0,\ i = 1, \dots, k$, coñécese como as *condicións de optimalidade de Karush-Kuhn-Tucker* (KKT) e indica que os $\hat{\alpha}_i$ se anulan cando as restricións $g_i(\hat{\mathbf{w}})$ non se saturan, isto é, cando $g_i(\hat{\mathbf{w}}) < 0$. Reciprocamente, os $g_i(\hat{\mathbf{w}})$ anúlanse cando os multiplicadores de Lagrange, $\hat{\alpha}_i$, son estritamente positivos.

Volvendo ao caso do clasificador de marxe máxima, temos que a función lagranxiana do problema de optimización (1.4) é

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle - \sum_{i=1}^{l} \alpha_i \left[y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) - 1\right], \tag{1.5}$$

sendo α_i os multiplicadores de Lagrange.

Para determinar o problema lagranxiano dual, necesitamos coñecer o ínfimo de $L(\mathbf{w}, b, \boldsymbol{\alpha})$, que podemos obter igualando a cero as derivadas parciais:

$$\begin{aligned}\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} =& \mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i = 0\\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} =& \sum_{i=1}^{l} \alpha_i y_i = 0.\end{aligned}$$

Nótese que da primeira ecuación deducimos que podemos expresar o vector perpendicular ao hiperplano, $\mathbf{w}$, como una combinación lineal dos exemplos de adestramento,

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i. \tag{1.6}$$

Combinando as ecuacións (1.5) e (1.6) obtemos

$$\begin{aligned}L(\mathbf{w}, b, \boldsymbol{\alpha}) =& \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle - \sum_{i=1}^{l} \alpha_i \left[y_i(\mathbf{w}, \mathbf{x}_i + b) - 1\right]\\ =& \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle - \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle + \sum_{i=1}^{l} \alpha_i\\ =& \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle,\end{aligned}$$

co que temos probado o seguinte resultado.

Proposición 1.3. *Sexan $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l))$ un conxunto de exemplos de adestramento linealmente separables e $\hat{\boldsymbol{\alpha}}$ un parámetro que é solución do seguinte problema de optimización cuadrática:*

$$\begin{aligned}\text{maximizar} \quad & W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle\\ \text{suxeito a} \quad & \sum_{i=1}^{l} y_i \alpha_i = 0,\\ & \alpha_i \geq 0, \quad i = 1, \dots, l.\end{aligned}$$

Entón, o vector de pesos $\hat{\mathbf{w}} = \sum_{i=1}^{l} y_i \hat{\alpha}_i \mathbf{x}_i$ correspóndese co hiperplano de marxe máxima.

Nótese que posto que o parámetro b non aparece no problema de optimización dual, temos que obter o seu valor empregando as restricións que tiñamos no problema primal, $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$. En principio, poderiamos resolver unha das l ecuacións e tomar $\hat{b}$ como a solución desta, porén, é máis correcto resolvelas todas e tomar o promedio ou ben empregar a expresión

$$\hat{b} = \frac{\text{mín}_{y_i=1}\{\langle \hat{\mathbf{w}}, \mathbf{x}_i \rangle\} - \text{máx}_{y_i=-1}\{\langle \hat{\mathbf{w}}, \mathbf{x}_i \rangle\}}{2}. \tag{1.7}$$

Analicemos agora as características da solución. Polas condicións de optimalidade de Karush-Kuhn-Tucker, que aparecen no Teorema 1.2, temos que a solución óptima dada por $\hat{\boldsymbol{\alpha}}$, $(\hat{\mathbf{w}}, \hat{b})$ debe satisfacer

$$\hat{\alpha}_i \left[y_i(\langle \hat{\mathbf{w}}, \mathbf{x}_i \rangle + \hat{b}) - 1 \right] = 0, \quad i = 1, \ldots, l.$$

En particular, temos que os únicos $\hat{\alpha}_i$ que non se anulan son aqueles que se corresponden con exemplos para os cales $y_i(\langle \hat{\mathbf{w}}, \mathbf{x}_i \rangle + \hat{b}) = 1$, é dicir, para aqueles que minimizan a distancia ao hiperplano. Como consecuencia disto, o vector de pesos, que vimos que podiamos expresar en función dos exemplos de adestramento, só depende en realidade dos datos que se atopan máis próximos ao hiperplano. A este conxunto de exemplos chamámolos *vectores de soporte* (ou *support vectors* na terminoloxía inglesa). Este nome xorde a raíz de que son soamente estes os datos que determinan ou "soportan" o hiperplano separador e os únicos para os cales pequenas alteracións poden afectarlle. Este subconxunto de exemplos será pois de gran importancia e denotamos o conxunto dos seus índices por sv. Temos por tanto que a ecuación que define o hiperplano separador vén dada por

$$H \equiv \sum_{i=1}^{l} y_i \hat{\alpha}_i \langle \mathbf{x}_i, \mathbf{x} \rangle + \hat{b} = \sum_{i \in sv} y_i \hat{\alpha}_i \langle \mathbf{x}_i, \mathbf{x} \rangle + \hat{b}.$$

Deste xeito, vemos como o valor dos multiplicadores de Lagrange nos dan unha interpretación intuitiva de canto de relevante é cada exemplo na obtención da solución. Por un lado, aqueles exemplos que non sexan support vectors non teñen influencia no resultado, co que a solución non se ve afectada aínda que estes sufran pequenas perturbacións. Por outro, os exemplos que si teñen un valor estritamente positivo para os multiplicadores de Lagrange, son máis relevantes canto máis grande é este valor.

Outra consecuencia importante que podemos deducir das condicións de KKT é que para os $i \in sv$,

$$y_i f(x_i) = y_i \left(\sum_{j \in sv} y_j \hat{\alpha}_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \hat{b} \right) = 1,$$

e por tanto

$$\langle \mathbf{w}, \mathbf{w} \rangle = \sum_{i,j=1}^{l} y_i y_j \hat{\alpha}_i \hat{\alpha}_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{j \in sv} \hat{\alpha}_j y_j \sum_{i \in sv} y_i \hat{\alpha}_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle$$
$$= \sum_{j \in sv} \hat{\alpha}_j (1 - \hat{b} y_j) = \sum_{j \in sv} \hat{\alpha}_j.$$

En consecuencia, a marxe do hiperplano óptimo podémola expresar como

$$\gamma = \frac{1}{\|\mathbf{w}\|} = \left(\sum_{i \in sv} \hat{\alpha}_i \right)^{-\frac{1}{2}}. \tag{1.8}$$

Vexamos agora un exemplo práctico desta versión máis simple dos SVM.

Exemplo 1.4. En primeiro lugar, simulamos un conxunto de datos en dúas dimensións linealmente separables $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l))$, que representamos na Figura 1.4

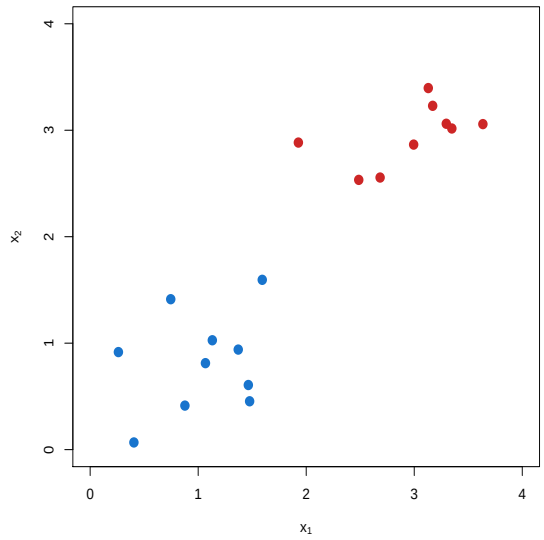


Figura 1.4: Datos simulados linealmente separables. A cor vermella correspóndese coa etiqueta positiva e a azul ca negativa.

Para obter o hiperplano separador empregamos a función `svm()` da librería `e1071` de R.

```
mod = svm(y ~ ., data = datos, scale = FALSE, kernel = "linear", cost = 10)
summary(mod)

...
## Number of Support Vectors:  3
##
##  ( 1 2 )
##
##
## Number of Classes:  2
##
## Levels:
##  -1 1
...
```

Os primeiros dous argumentos da función `svm()` indican que queremos clasificar os datos en función da etiqueta y tendo en conta todas as demais variables de `datos`, mentres que o terceiro, `scale=FALSE`, impide que se reescalen as entradas para que teñan media 0 e desviación típica 1. Dependendo dos datos que teñamos, ás veces será conveniente facer verdadeiro este parámetro. O resto dos argumentos discutirémolos con máis detalle no terceiro capítulo unha vez teñamos presentados outros algoritmos máis sofisticados.

Coa función `summary()` obtemos información básica do modelo. Omitimos parte da saída desta función, pois limítase a sinalar os argumentos que introducimos. Como vemos, amósase o número de support vectors e cantos hai en cada clase. Neste exemplo temos que hai tres support vectors dos cales un está na primeira clase e dous na segunda. A continuación aparecen o número de clases así como as súas correspondentes etiquetas.

Se ademais de saber o número de support vectors que hai os queremos identificar obtendo os seus índices, podémolo facer do seguinte xeito:

```
mod$index

## [1]  8 11 19
```

Para representar graficamente o modelo empregamos o comando `plot()`, co cal obtemos a Figura 1.5.

1.1.2. Transformacións a maior dimensión

Como xa comentamos ao inicio da sección anterior, os datos linealmente separables non son frecuentes na práctica e necesitamos por tanto refinar o algoritmo de aprendizaxe para

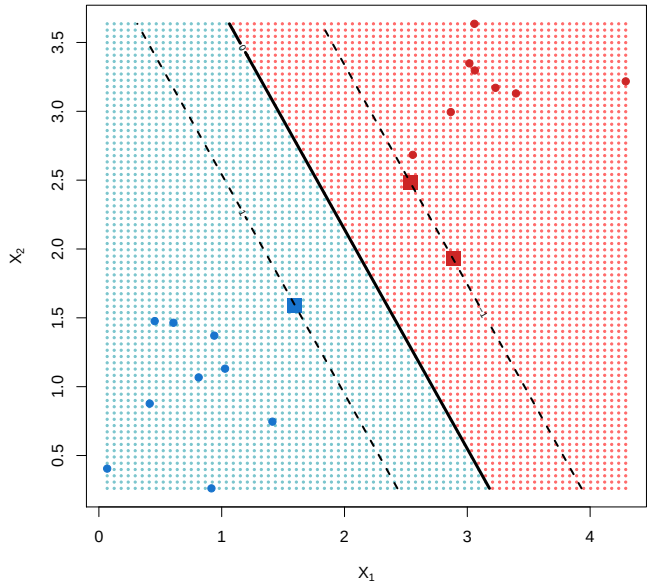


Figura 1.5: Facendo un `plot` do modelo podemos visualizar o hiperplano óptimo correspondente aos datos da Figura 1.4. A cor de cada rexión indica a regra de decisión que se emprega en cada punto. Se unha nova observación está na parte vermella do espazo, entón asignarémoslle unha etiqueta positiva, mentres que se está na parte azul, correspóndelle unha negativa. A liña onde se atopan ambas cores representa o hiperplano, no cal as novas observacións serían inclasificables. As liñas descontinuas a ambos lados do hiperplano son as marxes. Os exemplos máis próximos ao plano, os support vectors, aparecen representados cun punto cadrado de maior tamaño que o resto dos puntos, que se debuxan como círculos. Neste exemplo, podemos ver que existen un total de tres support vectors, sendo dous deles positivos e un negativo.

podelo empregar en casos máis xerais. É aquí onde aparece unha das ideas fundamentais dos SVM: a transformación dos datos a un espazo de dimensión maior no cal poidan ser linealmente separables. Ilustremos isto cun exemplo.

Exemplo 1.5. Consideremos o conxunto de datos $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l))$ representado na Figura 1.6. Claramente non podemos separar o conxunto de exemplos negativos dos positivos mediante unha recta. Neste caso, o problema de optimización que considerábamos anteriormente, (1.4), ten rexión factible baleira. En efecto, sexa unha recta arbitraria definida polo vector de pesos $\mathbf{w} \in \mathbb{R}$ e un escalar b e sexa $\mathbf{x}_i$ un exemplo mal clasificado segundo

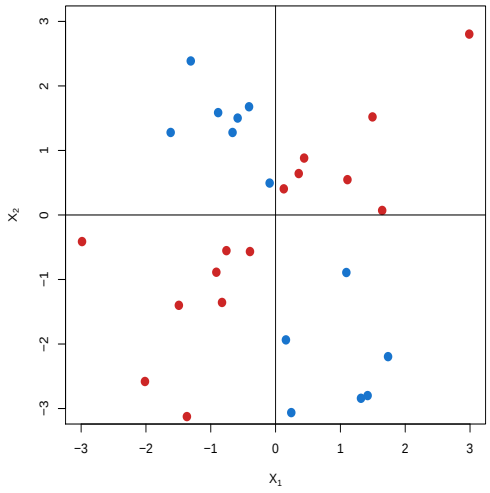


Figura 1.6: O conxunto de exemplos non é separable linealmente. Ademais, vese claramente que os datos positivos (en cor vermella) se atopan no primeiro e terceiro cuadrante, mentres que os negativos (en azul) están no segundo e cuarto.

a regra de decisión que induce esta recta (por ser o conxunto non linealmente separable sempre existirá polo menos un exemplo nestas condicións). Entón, $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) < 0$ logo $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 < 0$, co que non se verifican as restricións do problema de optimización (1.4). Así pois, o procedemento que empregamos na sección anterior non nos serve neste caso para atopar unha regra de decisión.

Agora ben, resulta evidente que os exemplos positivos se atopan no primeiro e terceiro cuadrante mentres que os negativos se concentran no segundo e no cuarto. Á vista desta información, podemos considerar a seguinte transformación

$$\begin{aligned} \phi : \mathbb{R}^2 &\longrightarrow \mathbb{R}^3 \\ (x_1, x_2) &\mapsto \phi(x_1, x_2) = (x_1, x_2, x_1 x_2) \end{aligned}$$

que fai que o conxunto de datos transformados, $\tilde{S} = ((\phi(\mathbf{x}_1), y_1), (\phi(x_2), y_2))$, sexa linealmente separable en $\mathbb{R}^3$. En particular, un plano que separa os datos é $\phi(\mathbf{x})_3 = 0$ como se pode apreciar na Figura 1.7.

Mediante esta transformación somos quen de separar os datos a costa de aumentar en un a dimensión de partida. De feito, dada a peculiar estrutura dos datos, poderiamos considerar unicamente a terceira coordenada, xa que é esta a que permite clasificar os datos, co cal non só non aumentariamos a dimensión senón que a reduciriamos. A pesar diso, eliminar as variables orixinais non adoita ser unha boa idea e os conxuntos de exem-

plos cos que traballamos en xeral non teñen unha estrutura tan sinxela, nin tan fácil de identificar a simple vista. Por isto, adoitaremos ter que realizar transformacións a espazos con dimensións moito maiores ca orixinal, mesmo podemos chegar a incluír infinitas variables. Por exemplo, dado un conxunto de datos en $\mathbb{R}^2$, podemos facer unha transformación $\phi : \mathbb{R}^2 \longrightarrow \mathbb{R}^\infty$ onde para cada $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$, a súa imaxe por ϕ vén dada por $\phi(\mathbf{x})_1 = x_1$, $\phi(\mathbf{x})_2 = x_2$, $\phi(\mathbf{x})_3 = x_1^2$, $\phi(\mathbf{x})_4 = x_1 x_2$, $\phi(\mathbf{x})_5 = x_2^2$, ...

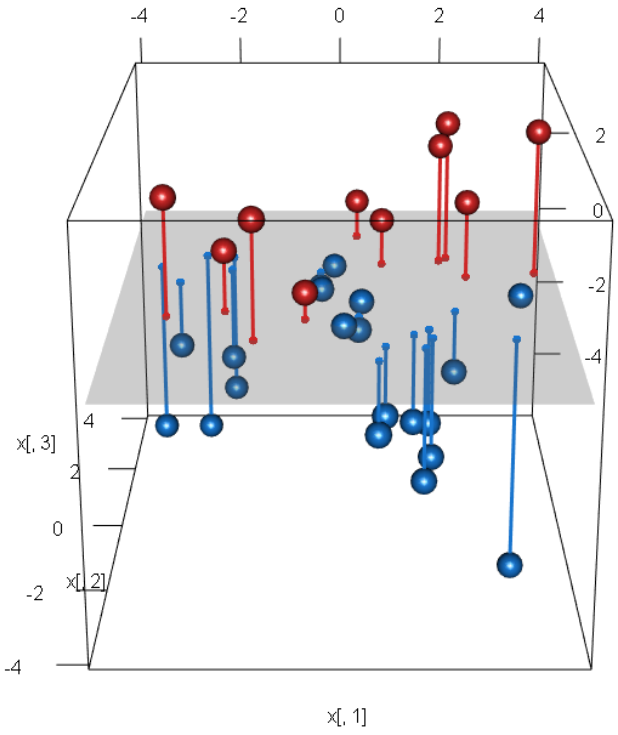


Figura 1.7: Ao transformar os datos segundo a aplicación ϕ, temos que os exemplos son agora separables polo plano $\phi(\mathbf{x})_3 = 0$, representado en gris, xa que as observacións positivas teñen a terceira coordenada maior que cero mentres que as negativas téñena menor que cero.

Os métodos kernel

Os métodos kernel aparecen como unha forma de evitar algunhas das limitacións que supón traballar con espazos de altas dimensións, como pode ser a imposibilidade de realizar certos cálculos nun tempo razoable, e consisten en definir a transformación a maior dimensión de forma implícita mediante os produtos interiores.

Definición 1.6. Chámase *función núcleo* ou *función kernel* a unha función K tal que

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle, \quad \forall \mathbf{x}, \mathbf{z} \in X$$

para algunha transformación $\phi : X \longrightarrow \mathcal{H}$.

Ademais, dado $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l))$ un conxunto de exemplos de adestramento, a matriz

$$\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{l}$$

coñécese como *matriz kernel.*

Se nos fixamos no problema de optimización descrito na Proposición 1.3 vemos que a función a optimizar, $\sum_{i=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$, só depende dos produtos interiores de vectores de exemplos, é dicir, de entradas da *matriz de Gram*, $G = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle)_{i,j=1}^{l}$. O mesmo ocorre se temos un novo dato $\mathbf{x}_0$ e o queremos clasificar, pois a regra de decisión ven dada polo signo de $f(\mathbf{x}_0) = \sum_{i=1}^{l} y_i \hat{\alpha}_i \langle \mathbf{x}_i, \mathbf{x}_0 \rangle + \hat{b}$ e, de novo, só emprega os produtos interiores entre $\mathbf{x}_0$ e os exemplos de adestramento. Así, dada unha función kernel que se corresponda cunha transformación ϕ dos datos, vemos que tanto para resolver o problema de optimización como para avaliar a regra de decisión no espazo transformado basta coñecer a matriz kernel. Non temos por tanto que coñecer a propia ϕ. Nen tan sequera as súas avaliacións nos vectores $\mathbf{x}1, \dots, \mathbf{x}_l$ e $\mathbf{x}_0$. De feito, na práctica o que se adoita facer é definir directamente a función kernel, co cal se define implicitamente o espazo transformado, en lugar de considerar o espazo transformado de partida e logo deducir cal tería que ser o seu produto interior. Non obstante, para poder facer isto, é necesario obter primeiro unhas condicións necesarias e suficientes que determinen cando unha función se corresponde co produto interior dalgún espazo.

Claramente, unha función kernel debe ser simétrica pois

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = \langle \phi(\mathbf{z}), \phi(\mathbf{x}) \rangle = K(\mathbf{z}, \mathbf{x})$$

e tamén debe satisfacer as desigualdades que se seguen da de Cauchy-Schwarz,

$$\begin{aligned} K(\mathbf{x}, \mathbf{z})^2 =& \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle^2 \leq \|\phi(\mathbf{x})\|^2 \|\phi(\mathbf{z})\|^2 \\ =& \langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle \langle \phi(\mathbf{z}), \phi(\mathbf{z}) \rangle = K(\mathbf{x}, \mathbf{x}) K(\mathbf{z}, \mathbf{z}). \end{aligned}$$

Porén, estas propiedades non son suficientes. Por exemplo, sexa $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ e supoñemos que a función $K(\mathbf{x}, \mathbf{z})$ é un kernel. Consideramos a matriz

$$\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{n}.$$

Como $\mathbf{K}$ é simétrica, existe unha matriz ortogonal $\mathbf{V}$ tal que $\mathbf{K} = \mathbf{V}\Lambda\mathbf{V}^T$, onde Λ é unha matriz diagonal contendo os autovalores de $\mathbf{K}$, λ_t, tales que os correspondentes autovectores, $\mathbf{v}_t = (v_{t1}, \dots, v_{tn})^T$, son as columnas de $\mathbf{V}$. Se supoñemos que todos os autovalores son non negativos, entón podemos considerar a aplicación

$$\phi : \mathbf{x}_i \mapsto \left(\sqrt{\lambda_1}v_{1i}, \dots, \sqrt{\lambda_n}v_{ni}\right)^T \in \mathbb{R}^n, i = 1, \dots, n.$$

Entón, temos que

$$\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \sum_{t=1}^{n} \lambda_t v_{ti} v_{tj} = (\mathbf{V}\Lambda\mathbf{V}^T)_{ij} = \mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j),$$

é dicir, $K(\mathbf{x}, \mathbf{z})$ é unha función kernel que se corresponde ca transformación ϕ.

Vexamos agora que a condición de ter todos os autovalores non negativos é necesaria. Sexa $\lambda_s < 0$ un autovalor con $\mathbf{v}_s$ o seu autovector asociado. Entón, se definimos o punto $\mathbf{z}$ como:

$$\mathbf{z} = \sum_{i=1}^{n} v_{si}\phi(\mathbf{x}_i) = \sqrt{\Lambda}\mathbf{V}^T\mathbf{v}_s.$$

Este tería como norma ao cadrado

$$\|\mathbf{z}\|^2 = \langle \mathbf{z}, \mathbf{z} \rangle = \mathbf{v}_s^T\mathbf{V}\sqrt{\Lambda}\sqrt{\Lambda}\mathbf{V}^T\mathbf{v}_s = \mathbf{v}_s^T\mathbf{K}\mathbf{v}_s = \lambda_s < 0,$$

o cal non é posible.

Por tanto, acabamos de probar o seguinte resultado.

Proposición 1.7. *Sexa $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ un espazo de entrada finito con $K(\mathbf{x}, \mathbf{z})$ unha función simétrica en X. Entón, $K(\mathbf{x}, \mathbf{z})$ é un kernel se e só se a matriz*

$$\boldsymbol{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{n}$$

é semidefinida positiva.

As condicións anteriormente definidas coñécense como *condicións de Mercer* e pódense xeneralizar a espazos de Hilbert de dimensión infinita no coñecido como Teorema de Mercer.[3]

Con estas observacións que acabamos de realizar, a demostración do seguinte resultado dedúcese inmediatamente da Proposición 1.3 e da ecuación (1.8).

Proposición 1.8. *Sexan $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l))$ un conxunto de exemplos de adestramento linealmente separable no espazo transformado definido implicitamente polo kernel*

[3] O Teorema de Mercer pódese consultar en Cristianini, N. and Shawe-Taylor, J. (2000) [2], páx. 35.

$K(\mathbf{x},\mathbf{z})$, $\mathcal{H}$, e supoñamos que os parámetros $\hat{\boldsymbol{\alpha}}$ e $\hat{b}$ son solución do problema de optimización cuadrática:

$$\begin{aligned} \text{maximizar} \quad & W(\boldsymbol{\alpha}) = \sum_{i,j=1}^{l} \alpha_i - \sum_{i=1}^{l} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{suxeito a} \quad & \sum_{i=1}^{l} y_i \alpha_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, \ldots, l. \end{aligned} \tag{1.9}$$

Entón, a regra de decisión dada polo signo de $f(\mathbf{x})$, onde

$$f(\mathbf{x}) = \sum_{i=1}^{l} y_i \hat{\alpha}_i K(\mathbf{x}_i, \mathbf{x}) + \hat{b},$$

é equivalente ao hiperplano de marxe máxima en $\mathcal{H}$, que ten marxe xeométrica

$$\gamma = \left(\sum_{i=1}^{l} \hat{\alpha}_i \right)^{\frac{1}{2}}.$$

Observación 1.9. Os requirimentos de que o kernel satisfaga as condicións de Mercer son equivalente a que a matriz de entradas

$$(K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{l}$$

sexa definida positiva para calquera conxunto de datos de adestramento. Isto implica á súa vez, que a matriz

$$(y_i y_j K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^{l}$$

é tamén definida positiva e por tanto, o problema de optimización dado por (1.9) é de optimización convexa. Grazas a isto, e como $\boldsymbol{\alpha} = 0$ é unha solución factible, garántase que o problema ten unha solución óptima única que podemos atopar de forma eficiente.

O uso de kernels proporciónanos unha vantaxe adicional pois a elección adecuada do kernel que empreguemos pode mellorar de forma moi significativa a capacidade de xeneralización. Aínda que existe un gran número de funcións kernels que podemos empregar, introduciremos tan só aquelas que son máis empregadas no eido das SVM.

Kernels Lineais Cando os exemplos de adestramento son linealmente separables no espazo de partida non necesitamos recorrer a unha transformación a un espazo de dimensión superior e empregamos o kernel

$$K(\mathbf{x},\mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle.$$

Kernels polinomiais Sexa $d \in \mathbb{N}$, definimos o *kernel polinomial de grao d* como

$$K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d .$$

O 1 añádese para que aparezan todos os termos de grao menor ou igual que d. De feito, podemos substituír este 1 por un parámetro θ para obter unha fórmula máis xeral.

Pódese comprobar que as funcións así definidas cumpren as condicións de Mercer e son por tanto kernels.[4] Por exemplo, se tomamos $d = 2$ e $l = 2$, o kernel polinomial viría dado por

$$\begin{aligned} K(\mathbf{x}, \mathbf{z}) =& 1 + 2x_1z_1 + 2x_2z_2 + 2x_1z_1x_2z_2 + x_1^2z_1^2 + x_2^2z_2^2 \\ =& \langle \phi(\mathbf{x}), \phi(\mathbf{x}) \rangle, \end{aligned}$$

para $\phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2x_2^2)$.

Este tipo de kernel adoita ofrecer bos resultados en datos normalizados.

Exemplo 1.10. Consideremos os puntos -1, 0, e 1 en $\mathbb{R}$ coas respectivas etiquetas 1, -1 e 1, representados na Figura 1.8. Claramente, estes datos non se poden separar linealmente no espazo de entrada $\mathbb{R}$. Empregando un kernel polinomial de grao 2, o problema dual sería

$$\begin{aligned} \text{maximizar} \quad & W(\boldsymbol{\alpha}) = \alpha_1 + \alpha_2 + \alpha_3 - (2\alpha_1^2 + \tfrac{1}{2}\alpha_2^2 + 2\alpha_3^2 - \alpha_2(\alpha_1 + \alpha_3)) \\ \text{suxeito a} \quad & \alpha_1 - \alpha_2 + \alpha_3 = 0, \\ & \alpha_i \geq 0, \quad i = 1, 2, 3. \end{aligned}$$

Da primeira restrición deducimos que $\alpha_2 = \alpha_1 + \alpha_3$ e, substituíndo na función a maximizar, temos que

$$W(\boldsymbol{\alpha}) = 2\alpha_1 + 2\alpha_3 - (2\alpha_1^2 - \tfrac{1}{2}(\alpha_1 + \alpha_3)^2 + 2\alpha_3^2).$$

Agora ben, como

$$\begin{aligned} \frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_1} =& 2 - 3\alpha_1 + \alpha_3 = 0 \\ \frac{\partial W(\boldsymbol{\alpha})}{\partial \alpha_3} =& 2 + \alpha_1 - 3\alpha_3 = 0 \end{aligned}$$

polas condicións de Karush-Kuhn-Tucker, temos que $\alpha_1 = \alpha_3 = 1$ e por tanto a solución óptima é

$$\hat{\alpha_1} = 1, \quad \hat{\alpha_2} = 2, \quad \hat{\alpha}_3 = 1$$

[4] Esta comprobación pódese ver en Abe, S. (2010) [1], páx. 459-561.

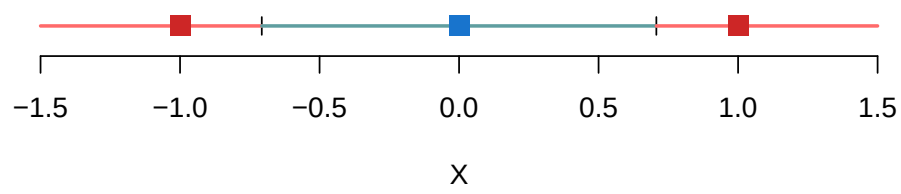


Figura 1.8: É claro que este conxunto de datos nunha dimensión non é separable linealmente. Se empregamos un kernel polinomial de grado 2 obtemos unha regra de decisión que clasifica correctamente os tres exemplos orixinais e que representamos con cores na recta X separadas por dous cortes verticais que representan a fronteira. Nótese que a distancia dos exemplos á fronteira non é a mesma en todos os casos, sendo esta considerablemente maior no dato negativo.

co que, neste caso, todos os exemplos son support vectors. Ademais, por (1.7), temos que

$$\begin{aligned} b =& \frac{\text{mín}_{y_i=1}\{\sum_{j=1}^{3} y_j y_i \hat{\alpha}_j \hat{\alpha}_i K(\mathbf{x}_j, \mathbf{x}_i)\} - \text{máx}_{y_i=-1}\{\sum_{j=1}^{3} y_j y_i \hat{\alpha}_j \hat{\alpha}_i K(\mathbf{x}_j, \mathbf{x}_i)\}}{2} \\ =& \frac{\text{mín}_{y_i=1}\{0\} - \text{máx}_{y_i=-1}\{2, 2\}}{2} = \frac{0-2}{2} = 1 \end{aligned}$$

logo a regra de decisión vén dada polo signo de

$$f(x_0) = K(-1, x_0) - 2K(0, x_0) + K(1, x_0) - 1 = (1 - x_0)^2 + (x_0 + 1)^2 - 3 = 2x_0^2 - 1.$$

É dicir, a fronteira de decisión vén dada por $x = \pm\sqrt{2}/2$. Na Figura 1.8 vemos que a distancia dos exemplos á fronteira é considerablemente maior no caso de x_2 ca nos dos outros dous exemplos pese a que a distancia dos tres datos ao hiperplano é a mesma no espazo transformado.

Kernels RBF Un dos kernels máis empregados debido aos bos resultados que adoita acadar son as *funcións kernel de base radial*, ou *kernels RBF* polas súas siglas en inglés, que definimos como:

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2),$$

onde γ é un parámetro positivo que controla o radio.

Nótese que a expresión anterior é equivalente a

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma}),$$

co que este kernel tamén recibe o nome de *kernel gaussiano* pola súa similitude coa función de distribución normal.

En xeral, podemos entender os kernels como unha medida da importancia que teñen os exemplos próximos á hora de clasificar un dato novo. Por exemplo, o kernel lineal asígnalle a todas as observacións a mesma importancia, sen ter en conta onde se atopan, mentres que o kernel de base radial asigna máis importancia aos datos máis próximos seguindo unha distribución normal. Deste xeito, é doado ver que o parámetro σ, ou de forma equivalente o γ, condiciona moito o desempeño que realiza o kernel na práctica, xa que é o que indica a varianza empregada. Se tomamos valores moi pequenos de σ entón para cuantificar a importancia das distancias estamos a empregar unha distribución normal con varianza moi pequena, é dicir, só estamos a ter en conta os datos moi próximos ao exemplo en cuestión, resultando o resto dos exemplos desprezables, co cal é moi probable que esteamos a cometer un sobreaxuste. Pola contra, canto máis grande sexa o valor de σ que tomamos, máis grande será a varianza da distribución empregada, co que se terán en conta datos cada vez máis afastados e se lles dará maior importancia, tendendo a comportarse coma un kernel lineal.

Na Figura 1.9 podemos ver a actuación dun kernel de base radial sobre un conxunto de datos empregando distintos valores do parámetro γ.

Kernel MLP O *kernel de perceptrón multicapa* ou, polas súas siglas en inglés, *kernel MLP* é tamén coñecido como *kernel sigmoide* pola súa similitude á función sigmoide empregada en regresión loxística. Defínese do seguinte xeito:

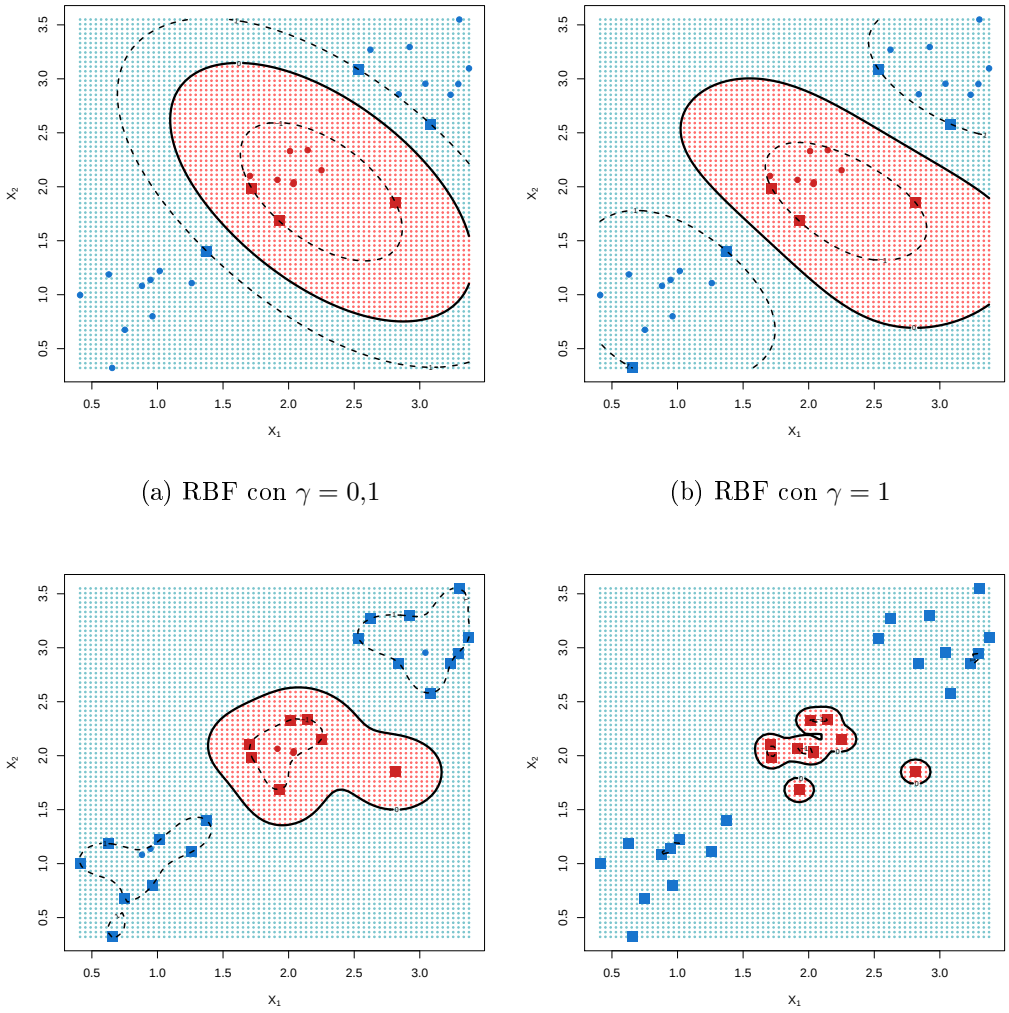
$$K(\mathbf{x}, \mathbf{z}) = \tanh(k\langle \mathbf{x}, \mathbf{z}\rangle + \theta),$$

onde k e θ son parámetros.

Este kernel é bastante popular debido a que a súa orixe está moi ligada á teoría de redes neuronais e adoita acadar bos resultados na práctica. Non obstante, só cumpre as hipóteses do Teorema de Mercer para certos parámetros, pero podemos garantir que isto ocorra tomando $k > 0$ e $\theta < 0$, sendo este último un valor pequeno. A pesar de que é un kernel que se pode aproximar polo RBF escollendo correctamente os parámetros, en particular se k é pequeno, para cantidades moi elevadas de datos é considerablemente máis rápido ca o kernel gaussiano, o cal explica a súa popularidade pese a que en xeral se comporte peor ca o kernel RBF.

1.1.3. Caso non separable

Pese a que o uso de kernels amplía en gran medida o número de problemas que podemos resolver co clasificador de marxe máxima, segue sen poder empregarse en moitos problemas



(a) RBF con $\gamma = 0{,}1$

(b) RBF con $\gamma = 1$

(c) RBF con $\gamma = 10$

(d) RBF con $\gamma = 100$

Figura 1.9: Nesta figura observamos o efecto de aplicar un kernel radial con distintos valores de γ a un conxunto de datos non separable linealmente. Como vemos, uns valores máis elevados de γ levan a ter en conta só os puntos máis próximos, chegando no caso da gráfica 1.9d a clasificar coa etiqueta negativa soamente os puntos inmediatamente máis próximos aos exemplos positivos. Nese caso, todos os exemplos son support vectors. Non obstante, con un γ máis pequeno o modelo ten un comportamento máis lineal, e no primeiro gráfico, o número de support vectors redúcese ata ter só seis. De seguir reducindo o valor de γ, chegaríamos a un problema de optimización con rexión factible baleira a non ser que os datos fosen linealmente separables.

reais, especialmente naqueles en que os datos presentan unha cantidade considerable de ruído. O feito de que obriguemos a que todos os exemplos estean ben clasificados implica que sempre temos unha hipótese perfectamente consistente, sen erro na aprendizaxe. Isto é así pois en caso contrario a marxe sería negativa, o cal non nos permite acoutar o erro cometido e o problema de optimización podería ter rexión factible baleira, e por tanto non haber solución.

Como comentabamos anteriormente, podemos entender os kernels como unha medida da importancia que se lle dá a cada exemplo dependendo da distancia á cal se atope do punto no que queremos efectuar a regra de decisión. Isto fai que en moitos casos sexa posible escoller parámetros que permitan que os exemplos de adestramento resulten separables no espazo transformado. Por exemplo, no caso do kernel gaussiano, podemos empregar un σ o suficientemente pequeno para que dado un exemplo, a contribución de todos os demais sexa insignificante. Isto levaríanos claramente a un sobreaxuste, pois identificariamos todo o espazo cunha das clases agás os puntos que se atopasen a unha ínfima distancia dos exemplos da outra clase. En consecuencia, forzar a que os datos resulten separables implica que temos un importante risco de que o modelo que obtemos se axuste en exceso ao conxunto de exemplos de adestramento e non realice unha boa xeneralización.

Vexamos cun exemplo como, pese a que o conxunto sexa separable, ás veces é conveniente clasificar mal algunha das observacións de partida para obter unha regra de decisión mellor. Consideremos os conxuntos de datos representados na Figura 1.10. Pese a que o conxunto é separable linealmente, a imposición de que todos os datos deben estar correctamente clasificados dá lugar a unha sensibilidade excesiva a certos exemplos, como comprobamos ao engadir un novo dato positivo moi preto dos negativos. Isto supón un gran problema porque agora hai varios datos que están moi preto do hiperplano, e como comentabamos ao principio do capítulo, canto máis lonxe se atope un dato da fronteira máis seguros podemos estar de que a clasificación que estamos a facer é correcta. Ademais, o feito de que o hiperplano varíe tanto engadindo un só dato parece indicar que estamos a realizar un sobreaxuste nos datos de adestramento.

Permitindo que algún dos datos resulte mal clasificado, podemos conseguir que o modelo sexa máis robusto ante observacións anómalas e obter unha maior xeneralización na maioría dos datos, é dicir, pese á clasificación incorrecta dalgún dos exemplos podemos ter máis seguridade na clasificación do resto dos datos. Por tanto, permitiremos que algunhas observacións estean entre a marxe e o hiperplano ou mesmo do lado incorrecto deste. Como se permite que algúns dos datos violen a marxe, este novo modelo denomínase *clasificador de marxe branda* ou *soft margin classifier*.

Para construír este novo modelo, partimos do problema de optimización que resolviamos

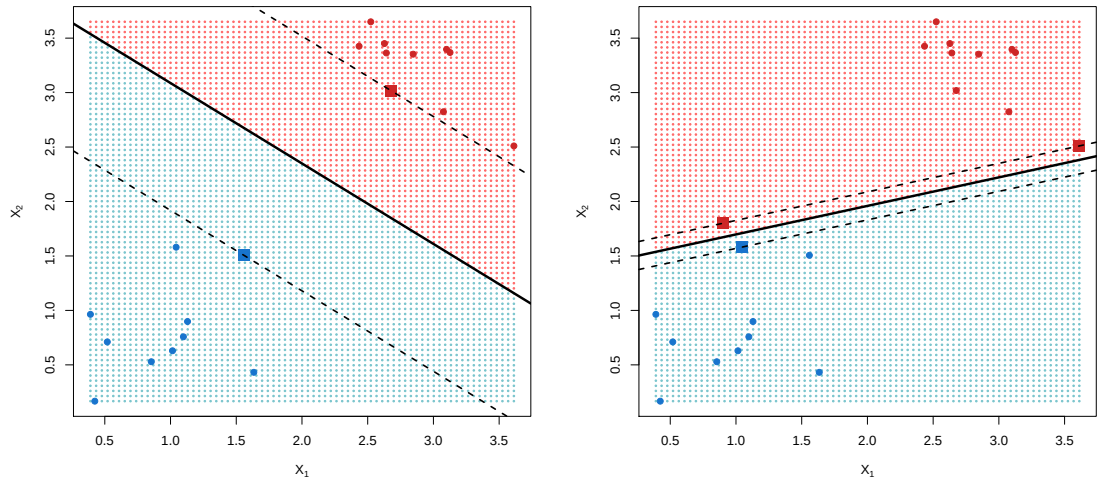


Figura 1.10: Na gráfica da esquerda temos un conxunto linealmente separable co seu hiperplano óptimo e as marxes correspondentes. Na imaxe da dereita engadimos un punto a maiores aos exemplos de adestramento. O conxunto resultante segue a ser separable linealmente e por tanto podemos calcular o seu hiperplano óptimo. Sen embargo, engadir un só punto fai que a marxe do clasificador (e por tanto a súa capacidade de xeneralización) varíe drasticamente.

no caso do clasificador de marxe máxima, lembremos:

$$\begin{aligned} &\text{minimizar} \quad \langle \mathbf{w}, \mathbf{w} \rangle \\ &\text{suxeito a} \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \quad \forall i = 1, \dots, l. \end{aligned}$$

para permitir o incumprimento da marxe introducimos unhas variables de relaxación positivas ξ_i, $i = 1, \dots, l$, que se denominan *variables slack*, co que as restricións serían agora

$$\begin{aligned} \text{suxeito a} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, l, \\ & \xi \geq 0 i = 1, \dots, l. \end{aligned}$$

Cómpre limitarmos o uso que facemos destas ξ_i, permitindo que só un número reducido de exemplos violen a marxe. En caso contrario, calquera hiperplano sería solución simplemente tomando uns valores de ξ_i o suficientemente grandes. Por esta razón, impoñemos unha penalización aos ξ_i con valores absolutos moi grandes engadindo un sumando na función a minimizar da forma $C \sum_{i=1}^{l} \xi_i^2$, sendo C unha constante positiva que debemos

determinar de antemán. O plano que buscamos é o que resolve o problema de optimización

$$\begin{aligned} \text{minimizar}_{\xi,\mathbf{w},b} \quad & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{l} \xi_i^2 \\ \text{suxeito a} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, l, \\ & \xi \geq 0, \quad i = 1, \dots, l. \end{aligned} \tag{1.10}$$

Na práctica, podemos escoller o parámetro C entre un amplo rango de valores. Canto maior é o valor de C, maior é custo de que un dato viole a marxe e por tanto tan só un número moi reducido de datos violarán a marxe e farano en xeral por moi pouca distancia. En consecuencia, a marxe será necesariamente máis estreita. Pola contra, se o valor de C é pequeno entón a clasificación incorrecta de exemplos non ten un custo elevado, polo que se permitirá que bastantes datos violen a marxe e podemos permitir que esta sexa máis ancha. A Figura 1.11 ilustra este comportamento.

Para escoller un valor óptimo, podemos calcular que parámetro dá mellores resultados nun conxunto de exemplos distinto ao de partida ou empregando técnicas de validación cruzada no propio conxunto de datos de adestramento. Isto é, que dividimos o conxunto de datos en dous grupos, empregando un deles para calcular o hiperplano óptimo para distintos valores de C e vendo cal destes valores permite unha mellor clasificación dos datos do segundo grupo, que non se empregaron para a obtención do modelo.

O problema de optimización (1.10) pódese simplificar eliminando as restricións de signo das variables slack. En efecto, supoñamos que $\xi_i < 0$ para algún i. Entón a primeira restrición,

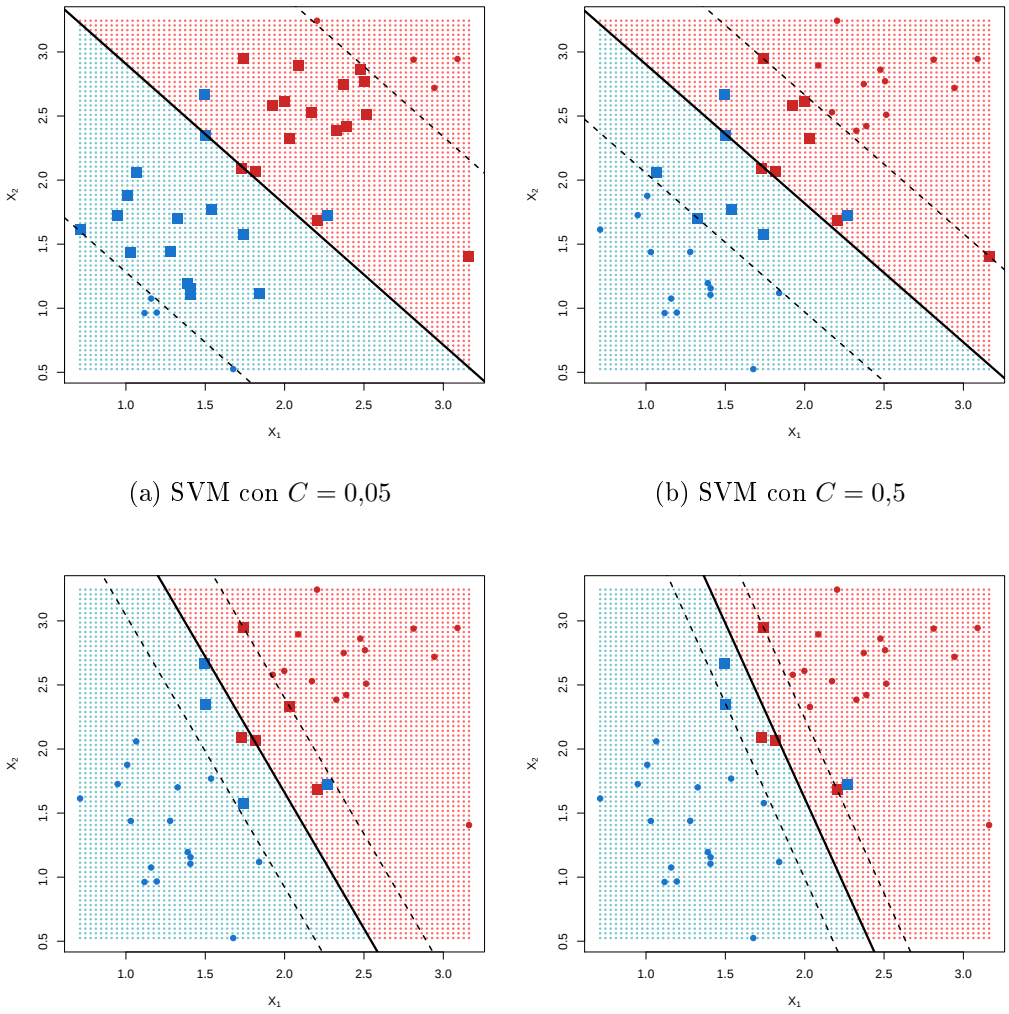
$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i,$$

cúmprese tamén para $\xi_i = 0$. Ademais, a función a minimizar é maior se a variable é distinta de cero que no caso $\xi_i = 0$. Por tanto, o problema resulta:

$$\begin{aligned} \text{minimizar}_{\xi,\mathbf{w},b} \quad & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{l} \xi_i^2 \\ \text{suxeito a} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \quad \forall i = 1, \dots, l. \end{aligned} \tag{1.11}$$

As variables slack proporciónannos información a cerca de onde se atopa o dato:

- Se $\xi_i = 0$ entón o dato está no lado correcto da marxe.
- Se $\xi_i > 0$ entón o exemplo está no lado incorrecto da marxe. Se ademais $\xi_i > 1$, entón o dato está mal clasificado.



(a) SVM con $C = 0{,}05$

(b) SVM con $C = 0{,}5$

(c) SVM con $C = 5$

(d) SVM con $C = 50$

Figura 1.11: Con valores máis pequenos de C reducimos o custo de violar a marxe, por iso no caso $C = 0{,}05$ vemos que a maioría dos datos se atopan entre a marxe e o hiperplano, ou mesmo mal clasificados. Temos por tanto un gran número de support vectors que se reduce de forma considerable ao aumentar o valor de C. Ademais, vemos que o hiperplano varía en función de C para acadar un maior número de datos ben clasificados a medida que o parámetro aumenta, pero sen chegar nunca a clasificalos todos correctamente pois o conxunto é non separable linealmente.

De novo, será máis sinxelo resolver o problema dual, polo que debemos calcular o lagranxiano do problema de optimización (1.11), que resulta

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i[y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) - 1 + \xi_i], \tag{1.12}$$

sendo os $\alpha_i, i = 1, \ldots, l$ os multiplicadores de Lagrange. Para obter a función a maximizar no problema dual temos que obter o ínfimo do lagranxiano, para o cal derivamos e igualamos a cero obtendo:

$$\begin{aligned}
\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{l} y_i\alpha_i\mathbf{x}_i = \mathbf{0} \\
\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial \boldsymbol{\xi}} &= C\boldsymbol{\xi} - \boldsymbol{\alpha} = \mathbf{0} \\
\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha})}{\partial b} &= \sum_{i=1}^{l} y_i\alpha_i = 0.
\end{aligned}$$

Nótese que en particular,

$$\mathbf{w} = \sum_{i=1}^{l} y_i\alpha_i\mathbf{x}_i \tag{1.13}$$

e

$$C\xi_i = \alpha_i, \quad i = 1, \ldots, l. \tag{1.14}$$

Tendo en conta isto e substituíndo en (1.12) temos

$$\begin{aligned}
L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) =& \frac{1}{2}\left\langle \sum_{j=1}^{l} y_j\alpha_j\mathbf{x}_j, \sum_{j=1}^{l} y_j\alpha_j\mathbf{x}_j \right\rangle + \frac{C}{2}\sum_{i=1}^{l}\xi_i^2 \\
&- \sum_{i=1}^{l}\alpha_i\left[y_i\left(\left\langle \sum_{j=1}^{l} y_j\alpha_j\mathbf{x}_j, \mathbf{x}_i \right\rangle + b \right) - 1 + \xi_i \right] \\
=& \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_iy_j\alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle + \frac{1}{2C}\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle - \frac{1}{C}\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle \\
=& \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_iy_j\alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle - \frac{1}{2C}\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle.
\end{aligned}$$

Agora ben, maximizar esta última expresión con respecto a $\boldsymbol{\alpha}$ equivale a maximizar

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_iy_j\alpha_i\alpha_j\left(\langle \mathbf{x}_i, \mathbf{x}_j\rangle + \frac{1}{C}\delta_{ij}\right),$$

sendo δ_{ij} a delta de Kronecker.

O problema dual resulta por tanto

$$\begin{aligned} \text{maximizar} \quad & W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left(\langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{1}{C} \delta_{ij} \right) \\ \text{suxeito a} \quad & \sum_{i=1}^{l} \alpha_i y_i = 0, \\ & \alpha_i \geq 0, \quad i = 1, \ldots, l. \end{aligned} \tag{1.15}$$

Ao igual que no caso separable, podemos empregar kernels pois tanto a función a optimizar no problema dual, $W(\boldsymbol{\alpha})$, coma as contas necesarias para clasificar un novo dato $\mathbf{x}_0$ dependen soamente dos produtos interiores entre os exemplos. Ao empregar kernels, o clasificador de marxe branda recibe o nome de *support vector machine*. En consecuencia, temos o seguinte resultado:

Proposición 1.11. *Dado $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l))$ un conxunto de exemplos de adestramento, que queremos clasificar nun espazo transformado definido implicitamente polo kernel $K(\mathbf{x}, \mathbf{z})$, supoñamos que os parámetros $\hat{\boldsymbol{\alpha}}$ son solución do problema de optimización cuadrática descrito en* (1.15).

Sexa $f(\mathbf{x}) = \sum_{i=1}^{l} y_i \hat{\alpha}_i K(\mathbf{x}_i, \mathbf{x}) + \hat{b}$, onde $\hat{b}$ escóllese verificando $y_i f(\mathbf{x}_i) = 1 - \hat{\alpha}_i / C$ para algún $i \in \{1, \ldots, l\}$ con $\hat{\alpha}_i \neq 0$. Entón, a regra de decisión dada polo signo de $f(\mathbf{x})$ é equivalente ao hiperplano no espazo transformado definido implicitamente por $K(\mathbf{x}, \mathbf{z})$, que resolve o problema de optimización (1.10)*, onde as variables slack se escollen en función da marxe xeométrica*

$$\gamma = \left(\sum_{i \in sv} \hat{\alpha}_i - \frac{1}{C} \langle \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\alpha}} \rangle \right)^{-\frac{1}{2}}.$$

Demostración. O hiperplano buscado está determinado polo vector de pesos $\hat{\mathbf{w}}$ e un certo escalar $\hat{b}$ que resolven (1.10). Xa argumentemos anteriormente que

$$\hat{\mathbf{w}} = \sum_{j=1}^{l} y_j \hat{\alpha}_j \mathbf{x}_j,$$

co que só nos falta calcular $\hat{b}$. Como este valor desaparece no problema dual, temos que obtelo directamente do primal empregando as condicións KKT, que veñen dadas por

$$\alpha_i [y_i (K(\mathbf{w}, \mathbf{x}_i) + b) - 1 + \xi_i] = 0.$$

Así, cando $\hat{\alpha}_i \neq 0$, temos unha ecuación na cal podemos despexar b, xa que

$$y_i \left(K \left(\sum_{j=1}^{l} y_j \hat{\alpha}_j \mathbf{x}_j, \mathbf{x}_i \right) + b \right) = 1 - \xi_i$$

e aplicando 1.14,

$$y_i \left(\sum_{j=1}^{l} y_j \hat{\alpha}_j K(\mathbf{x}_j, \mathbf{x}_i) + b \right) = 1 - \frac{\alpha_i}{C},$$

que coincide con $y_i f(\mathbf{x}_i) = 1 - \hat{\alpha}_i / C$ como queríamos probar.

Ademais, temos

$$\begin{aligned} \langle \hat{\mathbf{w}}, \hat{\mathbf{w}} \rangle =& K(\sum_{j=1}^{l} y_j \hat{\alpha}_j \mathbf{x}_j, \sum_{j=1}^{l} y_j \hat{\alpha}_j \mathbf{x}_j) = \sum_{i,j=1}^{l} y_i y_j \hat{\alpha}_i \hat{\alpha}_j K(\mathbf{x}_i, \mathbf{x}_j) \\ =& \sum_{j \in sv} \hat{\alpha}_j y_j \sum_{i \in sv} y_i \hat{\alpha}_i K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{j \in sv} \hat{\alpha}_j (1 - \frac{\alpha_i}{C} - y_i \hat{b}) \\ =& \sum_{j \in sv} \hat{\alpha}_j - \sum_{j \in sv} \hat{\alpha}_j \frac{\hat{\alpha}_i}{C} = \sum_{j \in sv} \hat{\alpha}_j - \frac{1}{C} \langle \hat{\alpha}_i, \hat{\alpha}_j \rangle. \end{aligned}$$

Logo a marxe do hiperplano buscado é

$$\gamma = \langle \hat{\mathbf{w}}, \hat{\mathbf{w}} \rangle^{-\frac{1}{2}} = \left(\sum_{i \in sv} \hat{\alpha}_i - \frac{1}{C} \langle \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\alpha}} \rangle \right)^{-\frac{1}{2}}.$$

□

Nótese que polas condicións KKT temos que se $\hat{\alpha}_i > 0$ entón, necesariamente

$$y_i (K(\hat{\mathbf{w}}, \mathbf{x}_i) + \hat{b}) - 1 + \xi_i = 0.$$

É dicir, que ou ben $y_i (K(\hat{\mathbf{w}}, \mathbf{x}_i) + \hat{b}) - 1 = 0$ ou $\xi_i > 0$. Isto indícanos que o exemplo $\mathbf{x}_i$ se atopa na marxe ou, pola contra, a viola. Agora ben, como tiñamos a relación $C\xi = \alpha_i$, temos que tan só os datos que violan a marxe teñen o parámetro $\hat{\alpha}_i$ positivo. Por tanto, se interpretamos de novo os multiplicadores de Lagrange como unha medición da importancia do exemplo correspondente á hora de determinar o hiperplano, temos que os exemplos que están correctamente clasificados e non violan a marxe, non contribúen á elección do hiperplano, mentres que si o fan os que a violan, incluídos os mal clasificados.

1.2. Clasificación multiclase

Ata agora, consideramos soamente a clasificación con dúas clases, é dicir, a clasificación binaria. Isto é debido a que a idea de separar os exemplos mediante un hiperplano non se xeneraliza facilmente ao caso no que temos máis de dúas clases. Porén, existen métodos para estender as SVM máis aló do caso binario. Consideraremos dúas das máis empregadas: a *clasificación un contra todos* e a *clasificación un contra un*.

1.2.1. Clasificación un contra todos

Supoñamos que temos un conxunto de exemplos divididos en k clases distintas, que denotaremos $1, \dots, k$. Para clasificar os datos mediante unha *support vector machine un contra todos*, consideramos k problemas de clasificación binaria separando unha clase das restantes. Para cada un de estes casos empregamos unha SVM que clasifique con etiqueta positiva os exemplos da clase i e con negativa o resto. Deste xeito temos (no espazo transformado) k hiperplanos separadores que dan lugar a k regras de decisión $f_i(\mathbf{x})$.

Para clasificar unha nova observación, consideramos o signo de

$$f_i(\mathbf{x}), \quad i = 1, \dots, k.$$

Se a observación verifica $f_i(\mathbf{x}) > 0$ para algún i, entón asígnaselle a clase i. A clasificación así definida é discreta, pois só se teñen en conta os signos das funcións de decisión.

Non obstante, a anterior clasificación dá lugar a rexións inclasificables no caso de que o signo de $f_i(\mathbf{x})$ sexa positivo para varios i ou de que non sexa positivo en ningún dos casos. Na Figura 1.12 ilústrase este comportamento.

Para evitar a existencia de rexións imposibles de clasificar empregamos funcións de decisión continuas en lugar de discretas. Para isto, en lugar de ter en conta soamente o signo das $f_i(\mathbf{x})$, consideramos tamén a súa magnitude, clasificando por tanto a observación $\mathbf{x}_0$ na clase na cal o valor de $f_i(\mathbf{x}_0)$ sexa maior. En consecuencia, os únicos puntos que resultan inclasificables son os aqueles para os cales existen i, j, con $i \neq j$ tales que $f_i(\mathbf{x}) = f_j(\mathbf{x})$ e este valor é o máximo das $f_k(\mathbf{x})$, é dicir, os puntos que se atopan nas fronteiras dos conxuntos. Na Figura 1.13 vemos o resultado de empregar funcións de decisión continuas en lugar de discretas cos datos da Figura 1.12.

1.2.2. Clasificación un contra un

No método da *clasificación un contra un* comparamos cada par de clases, resolvendo $k(k-1)/2$ problemas de clasificación. Como a cantidade de problemas a resolver é maior que no caso un contra todos adoitase realizar cada unha destas clasificacións sobre un subconxunto dos exemplos de cada clase involucrada.

Para determinar a clase dunha nova observación, realizamos as $k(k-1)/2$ regras de decisión e asignámoslle como clase aquela que se lle asignou con máis frecuencia. Formalmente, se denotamos por $f_{ij}(\mathbf{x})$ a regra de decisión entre as clases i e j, con etiquetas positiva e negativa respectivamente, temos que a regra de decisión entre j e i vén dada por

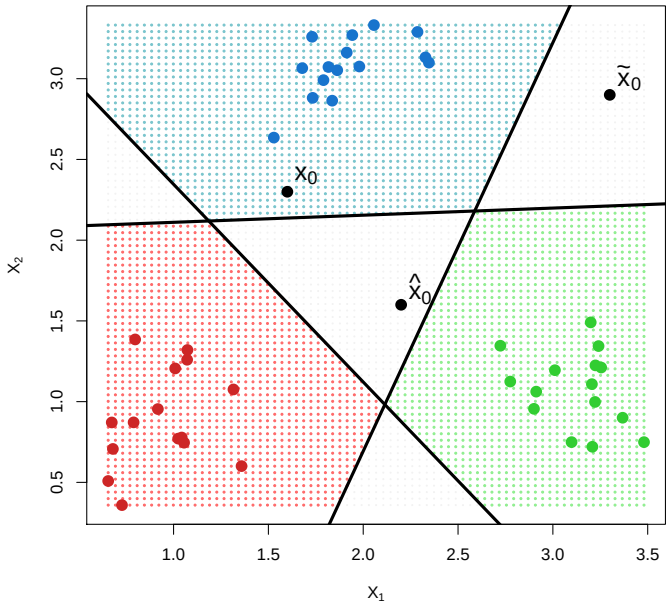


Figura 1.12: Na imaxe, temos representadas 3 clases distintas dun conxunto de exemplos. As liñas negras correspóndense cos hiperplanos separadores dos tres problemas de clasificación que separan cada clase das outras dúas. As rexións de cada cor representan os lugares onde a unha nova observación se lle asigna a clase da cor correspondente. Por exemplo, ao novo dato $\mathbf{x}_0$, asígnaselle a clase azul. Porén, existen rexións sen ningunha cor que indican os lugares onde os datos son inclasificables. A observación $\hat{\mathbf{x}}_0$ non é clasificable pois ningunha das regras de decisión $f_i(\hat{\mathbf{x}}_0)$ ten signo positivo. Pola contra, o dato $\tilde{\mathbf{x}}_0$ resulta non clasificable pois $f_i(\tilde{\mathbf{x}}_0) > 0$ tanto para a clase azul coma para a verde.

$f_{ji} = -f_{ij}$. Por tanto, as rexións do espazo

$$R_i = \{\mathbf{x} | f_{ij}(\mathbf{x}) > 0, j = 1, \dots, n, j \neq i\}, \quad i = 1, \dots, n,$$

é dicir, as rexións de puntos para os cales a regra de decisión $f_{ij}(\mathbf{x})$ é positiva para todo $j \neq i$, non se solapan. Claramente, aos puntos que se atopen nestas rexións asignaráselles a clase correspondente, pero non todos os puntos están contidos nalgunha das R_i. Para clasificar estes puntos definimos a función

$$f_i(\mathbf{x}) = \sum_{\substack{j=1 \\ i \neq j}}^{n} \text{sign}(f_{ij}(\mathbf{x})).$$

Entón, dada unha nova observación $\mathbf{x}_0$ asígnaselle, se existe, a clase para a cal $f_i(\mathbf{x}_0)$ sexa

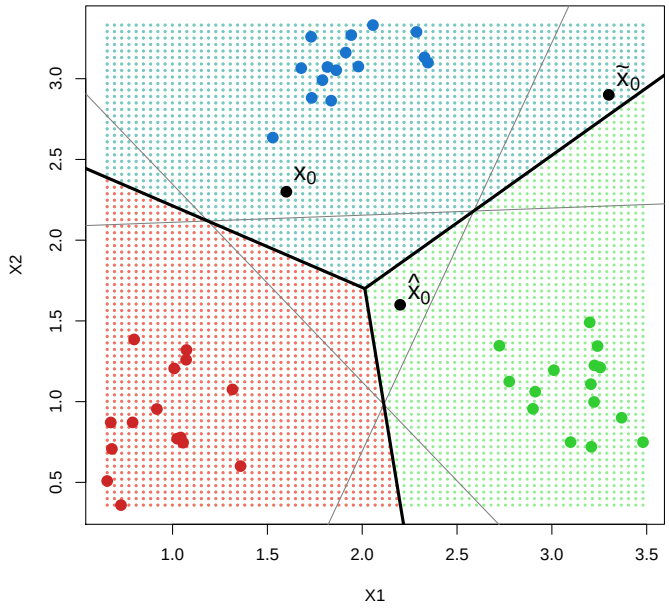


Figura 1.13: Empregando funcións de decisión continuas vemos que os únicos puntos inclasificables son aqueles que pertencen ás fronteiras que delimitan as rexións correspondentes con cada clase, co cal os exemplos $\hat{\mathbf{x}}_0$ e $\tilde{\mathbf{x}}_0$ son agora clasificables. As liñas de cor gris correspóndense cos tres hiperplanos separadores da figura anterior.

estritamente maior. Pese a isto, segue habendo rexións de puntos que resultan inclasificables, como se pode ver na Figura 1.14, aínda que estas serán polo xeral moito máis pequenas que no caso das SVM un contra todos discreta. Para evitar isto, podemos ter en conta as magnitudes dos $f_{ij}(\mathbf{x})$ en lugar de só o seu signo ao igual que faciamos no caso anterior, clasificando as novas observacións naquela clase i que maximice

$$\min_{\substack{j=1,\dots,k \\ j\neq i}} f_{ij}(\mathbf{x})$$

de entre aquelas con $f_i(\mathbf{x})$ máximo.

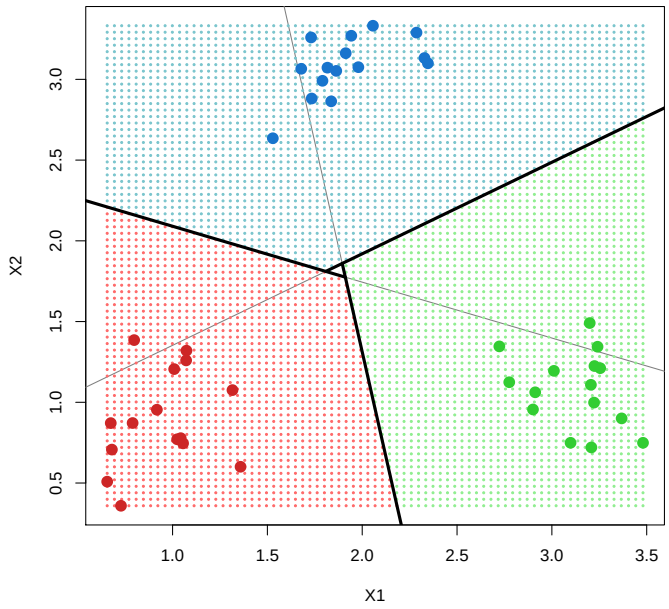


Figura 1.14: No método de clasificación multiclase un contra un, comparamos as clases dúas a dúas. Na imaxe observamos que para cada par de clases temos un hiperplano separador, que representamos cunha recta gris. As rexións coloreadas correspóndense cos conxuntos R_i das distintas clases, xa que neste caso, ao termos 3 clases, a única maioría posible é que ambas regras de decisión para unha clase sexan positivas. A fronteira destes conxuntos está resaltada cunha liña negra que coincide en parte cos hiperplanos separadores. Emporiso, existe unha pequena rexión sen cor no centro, na cal os datos resultan non clasificables. Isto débese a que os puntos desa rexión son clasificados a unha clase distinta por cada unha das regras de decisión. Esta rexión é considerablemente máis pequena que no caso da clasificación un contra todos discreta que observábamos na Figura 1.12.

Capítulo 2

SVM para a regresión

A metodoloxía das SVM que empregamos no capítulo anterior pódese modificar para poder aplicala ao caso da regresión. Recordemos que para a regresión queremos estimar o valor dunha variable resposta, Y, en función dunha variable explicativa, X. Suporemos que a relación entre estas variables ven dada por unha función $f(X)$. O noso obxectivo será por tanto a estimación desta función.

Supoñamos que temos un conxunto de exemplos da variable X, que denotaremos por $\mathbf{x}_1, \ldots, \mathbf{x}_l$. A cada un destes exemplos fáiselle corresponder un valor da variable Y, que poderá tomar agora calquera valor real, que denotaremos por $y_1, \ldots, y_l$ respectivamente.

Ao igual que na clasificación, comezaremos por introducir un caso sinxelo. Supoñemos por tanto que a función f que buscamos é lineal, isto é, que ven dada por un hiperplano

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

para certos parámetros $\mathbf{w}$ e b que debemos determinar. Agora ben, queremos escoller estes parámetros de forma que a función resultante axuste o mellor posible os datos. Para isto, debemos minimizar na medida do posible o *erro* ou *residuo* cometido en cada observación, $r = y_i - f(\mathbf{x}_i)$, que é a distancia á cal se atopa a función que eliximos con respecto ao valor real da variable Y nese punto. No modelo de regresión lineal tradicional, isto faise minimizando a suma de cadrados dos erros,

$$\sum_{i=1}^{l} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b)^2.$$

Non obstante, este método, ao elevar ao cadrado, fai que se teñan demasiado en conta os erros de maior tamaño, levando polo xeral a que os datos atípicos teñan unha gran influencia na solución.

Pola contra, na regresión mediante SVM, emprégase unha idea distinta que consiste en ignorar os erros cometidos se estes son o suficientemente pequenos e considerar o resto de forma lineal, seguindo a seguinte función:

$$E(r) = \begin{cases} 0, & \text{se } |r| \leq \varepsilon \\ |r| - \varepsilon, & \text{noutro caso,} \end{cases} \tag{2.1}$$

onde $\varepsilon > 0$ é unha constante pequena.

Na Figura 2.1 pódese apreciar a diferenza entre un modelo de regresión lineal tradicional e un empregando SVM.

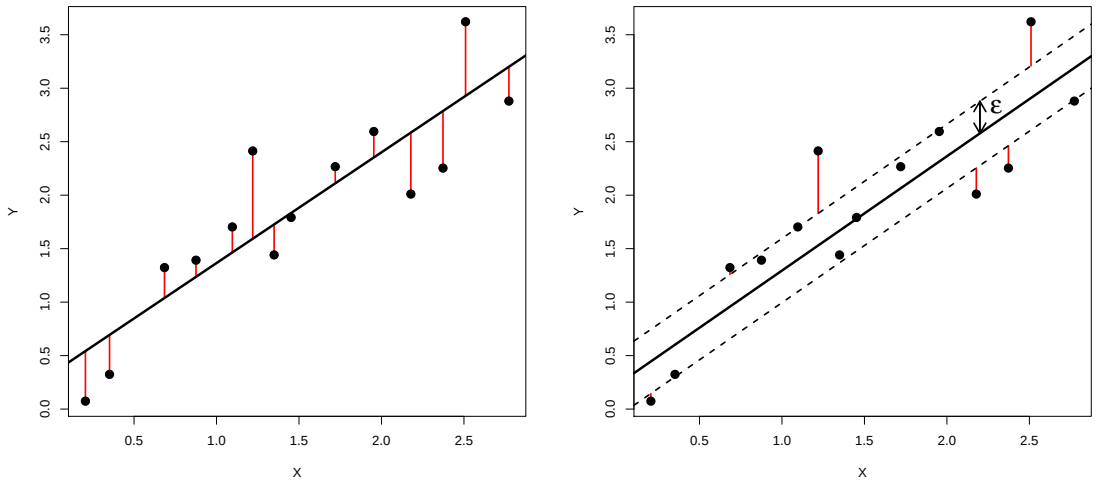


Figura 2.1: Na gráfica da esquerda vemos un conxunto de exemplos de dimensión un, xunto coa recta obtida mediante mínimos cadrados. Esta recta é a que minimiza a suma dos cadrados dos residuos, $r = y_i - wx_i + b$, que se representan mediante segmentos verticais de cor vermella. Na gráfica da dereita, no modelo proposto polo método das SVM, só se teñen en conta aqueles erros que superan en valor absoluto un certo valor ε. Graficamente podemos representar isto como aqueles exemplos que se atopan fóra dunha banda de grosor 2ε.

De (2.1) deducimos que o ideal sería que para todos os exemplos de adestramento o erro cometido fose tal que

$$|r| = |y - f(\mathbf{x})| \leq \varepsilon, \tag{2.2}$$

pois desta forma, todos os residuos computarían de forma nula. Non obstante, sen impoñer máis restricións calquera hiperplano serviría sen máis que tomar un ε o suficientemente

grande, co cal teríamos infinitas solucións posibles. Agora ben, isto non supón un problema pois o valor de ε está fixado de antemán.

Supoñamos que, fixado ε, existe polo menos un hiperplano para o cal todos os exemplos verifican (2.2). En xeral, pode darse o caso de que existan infinitos hiperplanos nestas condicións, e queremos atopar aquel que teña unha capacidade de xeneralización maior. Podemos asumir sen perda de xeneralidade que os datos que se atopan máis lonxe do hiperplano verifican $|y_i - f(\mathbf{x}_i)| = \varepsilon$, xa que se o residuo fose menor poderiamos tomar un ε máis pequeno e seguir verificando (2.2). Por tanto, definimos $\gamma = |y_i - f(\mathbf{x}_i)|$ como a *marxe* do hiperplano. Se maximizamos este valor, estamos a maximizar a probabilidade de que unha nova observación se atope nesta banda na cal aceptamos os erros, co cal considerariamos que o hiperplano axusta perfectamente a esta nova observación. Nótese que o valor de ε delimita o desfase que toleramos entre unha nova observación e o seu axuste antes de considerar que non resulta ben axustado.

Recordemos que como xa probamos en (1.3), a marxe vén dada por $\gamma = 1/\|\mathbf{w}\|$ e maximizar esta cantidade equivale a minimizar $\langle \mathbf{w}, \mathbf{w} \rangle$, co que o hiperplano buscado é o que resulta de resolver o problema de optimización

$$\begin{aligned} \text{minimizar} \quad & \langle \mathbf{w}, \mathbf{w} \rangle \\ \text{suxeito a} \quad & y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \varepsilon, \quad \forall i = 1, \dots, l, \\ & \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \varepsilon, \quad \forall i = 1, \dots, l. \end{aligned}$$

Ao igual que faciamos no caso do clasificador de marxe branda, podemos agora volver a introducir as variables slack para permitir exemplos con residuo $r > \varepsilon$ que non se atopen na banda que rodea $f(\mathbf{x})$. Denotamos por tanto

$$\xi_i = \begin{cases} 0, & \text{se } r - \varepsilon \leq 0 \\ r - \varepsilon, & \text{noutro caso} \end{cases}$$

$$\xi_i^* = \begin{cases} 0, & \text{se } r + \varepsilon \geq 0 \\ -r - \varepsilon, & \text{noutro caso} \end{cases},$$

como se ilustra na Figura 2.2.

O problema a resolver resulta agora

$$\begin{aligned} \text{minimizar} \quad & \langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{l} \left(\xi_i^2 + \xi_i^{*2} \right) \\ \text{suxeito a} \quad & y_i - (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \leq \varepsilon + \xi_i, \quad \forall i = 1, \dots, l, \\ & \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, \quad \forall i = 1, \dots, l, \\ & \xi_i \geq 0, \quad \xi_i^* \geq 0, \quad \forall i = 1, \dots, l. \end{aligned} \tag{2.3}$$

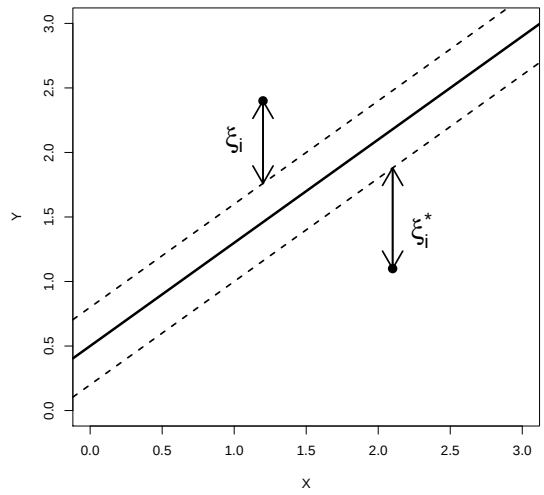


Figura 2.2: Introducimos as variables slack para permitir que existan exemplos que se atopen fóra da banda.

onde C é de novo unha constante positiva que escollemos de antemán e que penaliza o uso das variables slack. Para resolver este problema de optimización empregaremos outra vez o concepto da dualidade. Tense que o lagranxiano do anterior problema é

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*) =& \langle \mathbf{w}, \mathbf{w} \rangle + \sum_{i=1}^{l} (\xi_i^2 + \xi_i^{*2}) \\ & - \sum_{i=1}^{l} \alpha_i (\varepsilon + \xi_i - y_i + \langle \mathbf{w}, \mathbf{x}_i \rangle + b) \\ & - \sum_{i=1}^{l} \alpha_i^* (\varepsilon + \xi_i^* + y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b). \end{aligned} \tag{2.4}$$

Derivando (2.4) obtemos

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \mathbf{x}_i = \mathbf{0} \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*)}{\partial \boldsymbol{\xi}} &= C\boldsymbol{\xi} - \boldsymbol{\alpha} = \mathbf{0} \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*)}{\partial \boldsymbol{\xi}} &= C\boldsymbol{\xi}^* - \boldsymbol{\alpha}^* = \mathbf{0} \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*)}{\partial b} &= \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) = 0. \end{aligned} \tag{2.5}$$

En particular,

$$\mathbf{w} = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\mathbf{x}_i,$$

co que a función obxectivo resulta

$$f(\mathbf{x}) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\langle \mathbf{x}, \mathbf{x}_i\rangle + b. \tag{2.6}$$

Substituíndo (2.5) na expresión do lagranxiano, (2.4), obtemos

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*) =& \frac{1}{2}\left\langle \sum_{j=1}^{l}(\alpha_j - \alpha_j^*)\mathbf{x}_j, \sum_{j=1}^{l}(\alpha_j - \alpha_j^*)\mathbf{x}_j \right\rangle + \frac{C}{2}\sum_{i=1}^{l}(\xi_i^2 + \xi_i^*) \\ & - \sum_{i=1}^{l}\alpha_i\left(\varepsilon + \xi_i - y_i + \left\langle \sum_{j=1}^{l}(\alpha_j - \alpha_j^*)\mathbf{x}_j, \mathbf{x}_i \right\rangle + b\right) \\ & - \sum_{i=1}^{l}\alpha_i^*\left(\varepsilon + \xi_i^* + y_i - \left\langle \sum_{j=1}^{l}(\alpha_j - \alpha_j^*)\mathbf{x}_j, \mathbf{x}_i \right\rangle - b\right) = \\ =& \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \frac{1}{2C}\sum_{i=1}^{l}(\alpha_i^2 + \alpha_i^{*2}) \\ & - \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle \mathbf{x}_i, \mathbf{x}_j\rangle. \end{aligned}$$

Tal e como definimos as variables slack, resulta evidente que $\xi_i\xi_i^* = 0, \forall i = 1, \dots, l$. Ademais, vemos por (2.4) que podemos expresar α_i e α_i* en función de ξ_i e ξ_i^* respectivamente, co que se ten que $\alpha_i\alpha_i^* = 0, \forall i = 1, \dots, l$ e en consecuencia dáse a igualdade

$$\frac{1}{2C}\sum_{i=1}^{l}(\alpha_i^2 + \alpha_i^{*2}) = \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\frac{1}{C}\delta_{ij},$$

onde δ_{ij} é a delta de Kronecker.

Así, o problema dual resulta

$$\begin{aligned} \text{maximizar} \quad & W(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \\ & \qquad - \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(\langle \mathbf{x}_i, \mathbf{x}_j\rangle + \frac{1}{C}\delta_{ij}) \\ \text{suxeito a} \quad & \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0, \\ & \alpha_i \geq 0, \forall i = 1, \dots, l, \\ & \alpha_i^* \geq 0, \forall i = 1, \dots, l. \end{aligned} \tag{2.7}$$

cuxa solución debe verificar as condicións KKT:

$$\begin{aligned} \alpha_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b - y_i - \varepsilon - \xi_i) =& 0, \qquad \forall i = 1, \dots, l \\ \alpha_i^*(y_i - \langle \mathbf{w}, \mathbf{x}_i\rangle - b - \varepsilon - \xi_i^*) =& 0, \qquad \forall i = 1, \dots, l \\ C\xi_i =& \alpha_i, \qquad \forall i = 1, \dots, l \\ C\xi_i^* =& \alpha_i^*, \qquad \forall i = 1, \dots, l. \end{aligned} \tag{2.8}$$

Nótese que como consecuencia das dúas primeiras igualdades e tendo en conta que $\alpha_i\alpha_i^* = 0$, tense que ou ben $\alpha_i = 0$, $\alpha_i^* = 0$ ou se ten unha (e só unha) das seguintes igualdades:

$$\begin{aligned} \varepsilon = y_i - \langle \mathbf{w}, \mathbf{x}_i\rangle - b - \xi_i = 0, \quad & \text{se } \alpha_i > 0 \\ \varepsilon = \langle \mathbf{w}, \mathbf{x}_i\rangle + b - y_i - \xi_i^* = 0, \quad & \text{se } \alpha_i^* > 0 \end{aligned}$$

e por ser $C\xi_i = \alpha_i$ e $C\xi_i^* = \alpha_i^*$ temos que

$$|y_i - \langle \mathbf{w}, \mathbf{x}_i\rangle - b| = |y_i - f(\mathbf{x}_i)| > \varepsilon,$$

é dicir, que os exemplos para os cales α_i ou α_i^* son distintos de cero, os support vectors, son aqueles que se atopan fóra do tubo de radio ε que rodea o hiperplano.

Consideremos de novo o problema de optimización (2.7). Tanto nel coma na función obxectivo, que describiamos en (2.6), só empregamos as observacións por medio de produtos interiores. Entón, podemos volver a empregar kernels para ampliar o espazo de hipóteses que manexamos. Deste xeito, ao transformar os datos implicitamente mediante o kernel e efectuar a regresión mediante SVM no espazo transformado temos que a función obxectivo resultante ao desfacer a transformación xa non ten por que ser lineal.

Ademais, a similitude entre o problema de optimización que debemos resolver na regresión e o análogo para a clasificación é clara. Esta semellanza resulta aínda máis evidente se definimos unha nova variable $\boldsymbol{\beta} = \boldsymbol{\alpha} - \boldsymbol{\alpha}^*$ e empregamos a igualdade $\alpha_i\alpha_i^* = 0$ para reescribir (2.7) como:

$$\begin{aligned} \text{maximizar} \quad & W(\boldsymbol{\beta}) = \sum_{i=1}^{l} y_i\beta_i - \varepsilon\sum_{i=1}^{l}|\beta_i| - \frac{1}{2}\sum_{i,j=1}^{l}\beta_i\beta_j(\langle \mathbf{x}_i, \mathbf{x}_j\rangle + \frac{1}{C}\delta_{ij}) \\ \text{suxeito a} \quad & \sum_{i=1}^{l}\beta_i = 0. \end{aligned} \tag{2.9}$$

De feito, a única diferenza entre este problema e o (1.11) é que neste último as variables α_i están acoutadas inferiormente por cero, e isto non ocorre no caso das β_i.

Temos probado por tanto o seguinte resultado:

Proposición 2.1. *Sexa* $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l))$ *un conxunto de exemplos de adestramento que queremos axustar mediante un modelo de regresión. Sexa* $K(\mathbf{x}, \mathbf{z})$ *un kernel e supoñamos que os parámetros* $\hat{\boldsymbol{\beta}}$ *son solución do problema de optimización cuadrática*

$$\begin{aligned} \text{maximizar} \quad & W(\boldsymbol{\beta}) = \sum_{i=1}^{l} y_i \beta_i - \varepsilon \sum_{i=1}^{l} |\beta_i| - \frac{1}{2} \sum_{i,j=1}^{l} \beta_i \beta_j (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + \frac{1}{C} \delta_{ij}) \\ \text{suxeito a} \quad & \sum_{i=1}^{l} \beta_i = 0. \end{aligned} \tag{2.10}$$

Sexa $f(\mathbf{x}) = \sum_{i=1}^{l} \hat{\beta}_i K(\mathbf{x}_i, \mathbf{x}) + \hat{b}$*, onde* $\hat{b}$ *escóllese tal que* $f(\mathbf{x}_i) - y_i = -\varepsilon - \hat{\beta}_i / C$ *para calquera* i *tal que* $\hat{\beta}_i > 0$*. Entón, a función* $f(\mathbf{x})$ *é equivalente ao hiperplano do espazo transformado definido implicitamente polo kernel* $K(\mathbf{x}, \mathbf{z})$ *que resolve o problema de optimización* (2.3)*.*

Demostración. O único que falta probar é que a $\hat{b}$ se pode escoller da forma indicada, e isto é así pois se $\hat{\beta}_i = \hat{\alpha}_i - \hat{\alpha}_i^* > 0$ entón $\hat{\alpha}_i > 0$. Polas condicións de optimalidade KKT, que enunciamos en (2.8), e empregando $C\boldsymbol{\xi} = \boldsymbol{\alpha}$ temos que

$$\hat{b} = y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - \varepsilon - \frac{1}{C} \hat{\alpha}_i.$$

que equivale á ecuación

$$f(\mathbf{x}_i) - y_i = -\varepsilon - \frac{1}{C} \hat{\beta}_i$$

como queriamos demostrar. □

Capítulo 3

Implementación en R

3.1. Clasificación

Empregaremos principalmente a librería de R `e1071`, aínda que existen outras librerías que se poden empregar para resolver SVM como `kernlab` ou `klaR`.

En `e1071` temos a función `svm()`, que se usa para resolver todo tipo de SVM, dende o clasificador de marxe máximo ata o clasificador de marxe branda ou o modelo de regresión. Para ilustrar o seu uso empregaremos o conxunto de datos Blood Transfusion Service Center Data Set empregado nun artigo de Yeh. I e outros [9], que se pode atopar en UCI Machine Learinig Repository [3], no cal recóllese información das doazóns de sangue realizadas por un conxunto de 748 individuos. A variable obxecto de interese indica se o individuo en cuestión realizou unha doazón de sangue no último mes, que denotaremos por 1 se fixo a doazón e 0 en caso contrario. Para realizar a clasificación desta variable teremos en conta as seguintes:

- Recency: indica o número de meses que transcorreron dende a última doazón.
- Frecuency: correspóndese co número total de doazóns realizadas.
- TotalDonated: indica a cantidade total de sangue doado medida en centímetros cúbicos.
- Time: é o tempo, en meses, transcorrido dende a primeira doazón realizada polo individuo.

Para cargar os datos empregamos o comando

```
datos <- read.table("transfusion.data", header = TRUE, sep = ",")
names(datos) <- c("Recency", "Frecuency", "TotalDonated", "Time", "HasDonated")
```

Como queremos empregar un modelo de clasificación, debemos converter a variable obxectivo nun factor,

```
datos$HasDonated <- as.factor(datos$HasDonated)
```

pois en caso contrario, ao tratarse dunha variable numérica, a función `svm()` empregaría por defecto o modelo de regresión.

Podemos ver agora un resumo dos datos empregando `summary()`.

```
summary(datos)
```

```
##     Recency         Frecuency       TotalDonated        Time
##  Min.   : 0.000   Min.   : 1.000   Min.   :  250   Min.   : 2.00
##  1st Qu.: 2.750   1st Qu.: 2.000   1st Qu.:  500   1st Qu.:16.00
##  Median : 7.000   Median : 4.000   Median : 1000   Median :28.00
##  Mean   : 9.507   Mean   : 5.515   Mean   : 1379   Mean   :34.28
##  3rd Qu.:14.000   3rd Qu.: 7.000   3rd Qu.: 1750   3rd Qu.:50.00
##  Max.   :74.000   Max.   :50.000   Max.   :12500   Max.   :98.00
##  HasDonated
##  0:570
##  1:178
##
##
##
##
```

Ademais, podemos representar graficamente os datos comparando as variables dúas a dúas mediante o comando `pairs()`, que da lugar á gráfica representada na Figura 3.1. Como vemos, as variables `Frecuency` e `TotalDonated` son completamente proporcionais, polo que podemos realizar o modelo sen ter en conta esta última pois ambas variables aportan a mesma información. En consecuencia, suprimiremos esta variable do conxunto de datos.

```
datos <- datos[, c(1, 2, 4, 5)]
```

Para ver a capacidade de xeneralización das distintas SVM que empreguemos escolleremos un subconxunto de datos que serán os que empreguemos para o adestramento do algoritmo e posteriormente empregaremos as regras de decisión obtidas no resto dos datos para comprobar a capacidade de xeneralización. Para escoller o conxunto de exemplos

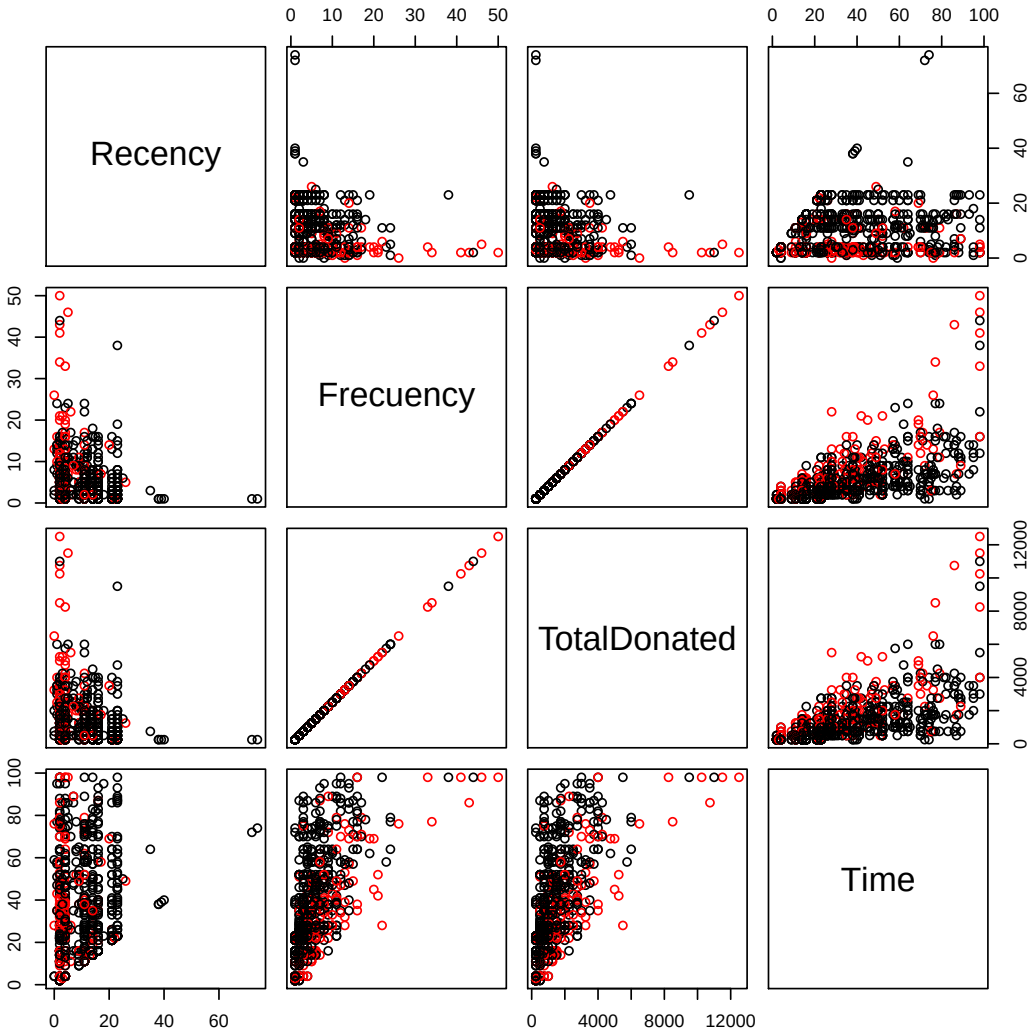


Figura 3.1: O comando `pairs()` representa cada par de variables enfrontadas. Ademais, a clase de cada observación indícase mediante a súa cor, sendo a cor vermella a que se corresponde cos doadores do último mes.

de adestramento seleccionamos de forma aleatoria 560 exemplos de entre todos os datos dispoñibles, o cal supón aproximadamente o 75 % do total das observacións.

```
set.seed(1)
n = nrow(datos)
k = 560
x <- sort(sample(1:n, k))
```

```
test <- datos[x, ]
```

Consideramos en primeiro lugar un kernel lineal:

```
mod = svm(HasDonated ~ ., data = test, kernel = "linear")
summary(mod)

...
## Number of Support Vectors:  278
##
##  ( 138 140 )
##
##
## Number of Classes:  2
##
## Levels:
##  0 1
...
```

Con esta notación indicamos que se empreguen os exemplos do data.frame `test` e que se clasifique a variable `HasDonated` en relación ao resto. Por defecto, a función `svm()` asigna un kernel de base radial, polo que temos que especificar que o kernel empregado sexa lineal. Ademais, o valor de C por defecto é un, pero podémolo cambiar especificando o argumento `cost`. Con este modelo podemos realizar unha predición sobre o resto dos datos

```
pred <- predict(mod, datos[-x, ])
```

co que obtemos os seguintes resultados

```
table(pred, true = datos$HasDonated[-x])

##      true
## pred   0   1
##    0 146  37
##    1   4   1
```

Vemos así que se clasifican incorrectamente un total de 41 dos 188 datos restantes, o cal supón unha porcentaxe de acerto do 78.2 %. Non obstante, estamos a clasificar incorrectamente todos os datos que orixinalmente tiñan etiqueta 1 agás un, o cal non é un bo resultado. Isto pode deberse a que o número de doadores no último mes é tan só unha pequena parte do total de individuos. Para evitar que o erro cometido á hora de clasificar a individuos coa etiqueta 1 como non doadores podemos outorgarlle un peso maior a esta etiqueta. Isto podémolo facer engadindo o argumento `class.weights` á chamada do SVM.

```
mod = svm(HasDonated ~ ., data = test, class.weights = c(`0` = 1, `1` = 3),
    kernel = "linear")
pred <- predict(mod, datos[-x, ])
table(pred, true = datos$HasDonated[-x])

##      true
## pred  0  1
##    0 93 11
##    1 57 27
```

Así, obtemos unha maior porcentaxe de datos ben clasificados de entre aqueles que teñen etiqueta 1 pero a cambio reducimos a porcentaxe total de acertos ao 63.8 %. Pese a que isto pode parecen contraproducente, dependendo do noso obxectivo ten moita utilidade. Por exemplo, se a variable obxectivo é a presenza ou non dunha enfermidade é preferible diagnosticar erroneamente a unha persoa que non padeza dita enfermidade que clasificar como sa a unha persoa enferma.

Agora ben, como comentabamos no primeiro capítulo, a elección dos parámetros adoita xogar un papel importante á hora de determinar a capacidade de xeneralización dunha SVM. Por esta razón, é aconsellable probar distintos valores de C. Isto podémolo facer empregando a función `tune()`, que utiliza técnicas de validación cruzada para obter o valor óptimo de entre os valores que lle proporcionamos.

```
tune.out = tune(svm, HasDonated ~ ., data = test, kernel = "radial",
    ranges = list(cost = c(0.001, 0.01, 0.1, 1, 10, 100)),
    class.weights = c(`0` = 1, `1` = 3))
summary(tune.out)

...
## - best parameters:
##  cost
##    10
##
## - best performance: 0.3142857
##
## - Detailed performance results:
##    cost     error dispersion
## 1 1e-03 0.5571429 0.26483748
## 2 1e-02 0.4250000 0.16033907
## 3 1e-01 0.3464286 0.03388155
...
```

Vemos por tanto que o valor óptimo de C para o conxunto de datos de adestramento é $C = 10$. Para obter o modelo co parámetro óptimo basta empregar a variable `best.model` de `tune.out`, co que obtemos os seguintes erros na predición:

```
mod = tune.out$best.model
pred <- predict(mod, datos[-x, ])
table(pred, tipo = datos$HasDonated[-x])

##      tipo
## pred   0   1
##    0 105  14
##    1  45  24
```

Como vemos, cometemos agora menos erros que con $C = 1$, o que dá lugar a unha correcta clasificación do 68.6 % das observacións. Agora ben, o kernel lineal que empregamos ata agora adoita dar resultados pobres con estruturas de datos máis complexas, co que é recomendable probar con outros kernels. A función `svm()` permite empregar os kernels lineal, polinómico, gaussiano e sigmoide que describiamos no primeiro capítulo. Para usalos, basta substituír `kernel="linear"` por `kernel="radial"`, `kernel="polynomial"` ou `kernel="sigmoid"` respectivamente na chamada á función `svm()`. Por exemplo, podemos empregar un kernel gaussiano cun valor de $\gamma = 1$ mediante o comando

```
mod = svm(HasDonated ~ ., data = test, kernel = "radial", gamma = 1)
```

De novo podemos empregar a función `tune()` para obter os parámetros óptimos.

```
tune.out = tune(svm, HasDonated ~ ., data = test, kernel = "radial",
    ranges = list(cost = c(0.001, 0.01, 0.1, 1,10,100),
         gamma = c(0.1 ,1 ,10, 100)),
    class.weights = c("0" = 1, "1" = 3))
summary(tune.out)

...
## - best parameters:
##  cost gamma
##     1   100
##
## - best performance: 0.3125
##
## - Detailed performance results:
##     cost gamma     error dispersion
## 1  1e-03   0.1 0.4267857 0.22859778
...
```

Temos por tanto que os parámetros óptimos para este kernel son $C = 1$ e $\gamma = 100$. Co que obtemos os seguintes resultados para a xeneralización

```
mod = tune.out$best.model
pred <- predict(mod, datos[-x, ])
table(pred, tipo = datos$HasDonated[-x])

##      tipo
## pred   0   1
##    0 136  27
##    1  14  11
```

o que supón unha clasificación correcta do 78.2 % dos datos, que é unha mellora considerable con respecto ao kernel lineal.

Se queremos empregar o kernel polinomial, debemos escoller o seu grao así como o parámetro θ, que se denotan `degree` e `coef0` na función `svm()`. Por exemplo, para empregar un kernel polinomial de grao 3 e con $\theta = 1$ usamos o comando

```
mod = svm(HasDonated ~ ., data = test, kernel = "polynomial", degree = 3,
    coef0 = 1)
```

Para empregar un kernel sigmoide, escollemos os parámetros k e θ que veñen dados por `gamma` e `coef0`. Aínda que a función `svm()` permite escoller como `coef0` calquera valor real, só podemos asegurar que o kernel está ben definido no caso en que este valor sexa negativo.

Vexamos agora como obter información importante a cerca do modelo. Para saber o número de support vectors que se empregan en total usamos o comando `mod$tot.nSV`, mentres que se o que queremos é coñecer a cantidade de support vector de cada clase empregamos `mod$nSV`. Para obter os exemplos que se corresponden cos support vectors podemos usar `mod$SV` mais é posible que estes estean reescalados. Para evitar isto podemos obter os seus índices mediante `mod$index` e obtelos directamente da matriz de datos. Ademais, podemos obter os $y_i\alpha_i$, $\forall i = 1, \dots, l$ con `mod$coefs` e $-b$ vén dado por `mod$rho`.

3.2. Regresión

Para ilustrar o uso das SVM para a regresión empregamos datos relacionados coa educación de distintos países obtidos de The World Bank e orixinarios das bases de datos da Unesco. O obxectivo será agora buscar unha relación entre a renda nacional bruta de cada país e as seguintes variables:

- Gasto do Goberno en educación (en porcentaxe do PIB).
- Porcentaxe de poboación que completou os estudos primarios.
- Porcentaxe de matriculados no primeiro ciclo dos estudos secundarios.
- Porcentaxe de matriculados no segundo ciclo dos estudos secundarios.
- Porcentaxe de matriculados en estudios medios.
- Porcentaxe de matriculados en estudos superiores.

Estas porcentaxes de matriculados están calculadas sobre o total da poboación en idade de realizar ditos estudos. Vexamos un resumo dos datos:

```
summary(datos)
```

```
##     Gov.exp         Prim.compl       ER.LowSec        ER.Prim
##  Min.   : 1.511   Min.   : 37.83   Min.   :12.02   Min.   : 41.44
##  1st Qu.: 3.283   1st Qu.: 86.73   1st Qu.:58.86   1st Qu.: 90.11
##  Median : 4.579   Median : 97.34   Median :85.21   Median : 95.72
##  Mean   : 4.830   Mean   : 91.82   Mean   :73.93   Mean   : 91.69
##  3rd Qu.: 5.814   3rd Qu.:101.22   3rd Qu.:93.61   3rd Qu.: 98.30
##  Max.   :13.125   Max.   :124.11   Max.   :99.75   Max.   :100.00
##     ER.UpSec        ER.postSec            ER.Ter            GNI
##  Min.   : 1.939   Min.   :  0.03913   Min.   : 30.57   Min.   :   320
##  1st Qu.:29.177   1st Qu.:  1.41627   1st Qu.: 67.08   1st Qu.:  1840
##  Median :63.118   Median :  5.30795   Median : 77.91   Median :  5740
##  Mean   :56.200   Mean   : 14.63483   Mean   : 77.95   Mean   : 14461
##  3rd Qu.:82.142   3rd Qu.: 22.41575   3rd Qu.: 90.45   3rd Qu.: 16050
##  Max.   :97.078   Max.   :123.01578   Max.   :115.74   Max.   :116300
```

Podemos observar que existen algunhas porcentaxes superiores ao 100 %. Isto é debido a que a poboación que cursa certos estudos pode non estar contida por completo no grupo de idades que se lle asigna. Ademais, hai países onde non son obrigatorios os rexistros dos nacementos, co cal estas porcentaxes son aínda menos fiables.

De novo, empregamos como datos de adestramento o 75 % dos datos. A mesma función que empregamos para a clasificación válenos agora para realizar a regresión e tamén podemos axustar os parámetros que desexemos coa función `tune()`. Por exemplo, podemos realizar un axuste mediante kernel lineal e buscar cales son os valores óptimos para o custo e o parámetro ε.

```
tune.out = tune(svm, GNI ~ ., data = test, kernel = "linear",
    ranges = list(cost = c(0.01, 0.1, 1,10, 100),
        epsilon = c(0.01, 0.05, 0.1)))
summary(tune.out)

...
## - best parameters:
##  cost epsilon
##   100     0.1
##
## - best performance: 385935745
...

mod<-tune.out$best.model
```

Para ver o erro cometido empregamos os residuos do modelo, $|y_i - f(\mathbf{x}_i)|$ e calculamos a súa media e comparámola coa media da renda nacional bruta, que é a mellor predición que podemos facer sen ter en conta o resto das variables.

```
mean(abs(datos$GNI[-x] - predict(mod, datos[-x, ])))

## [1] 9932.164

mean(abs(datos$GNI[-x] - mean(datos$GNI[-x])))

## [1] 13986.1
```

Como vemos, a mellora é evidente, mais se facemos unha representación dos datos, que se pode ver na Figura 3.2, vemos que axustar a variable `GNI` mediante un hiperplano non parece o máis adecuado. Por tanto, probaremos distintos kernels para ver se conseguimos un axuste mellor.

Se realizamos o proceso anterior cos kernels polinomial, radial e sigmoide obtemos os seguintes resultados:

```
##         Lineal Polinomial Radial Sigmoide
## Erro      9932       3977   7269    12795
## Mellora   4054      10009   6717     1191
```

Con esta gráfica vemos que o mellor axuste é o que se corresponde co kernel gaussiano, que dá lugar a erros de media moito menores que os demais.

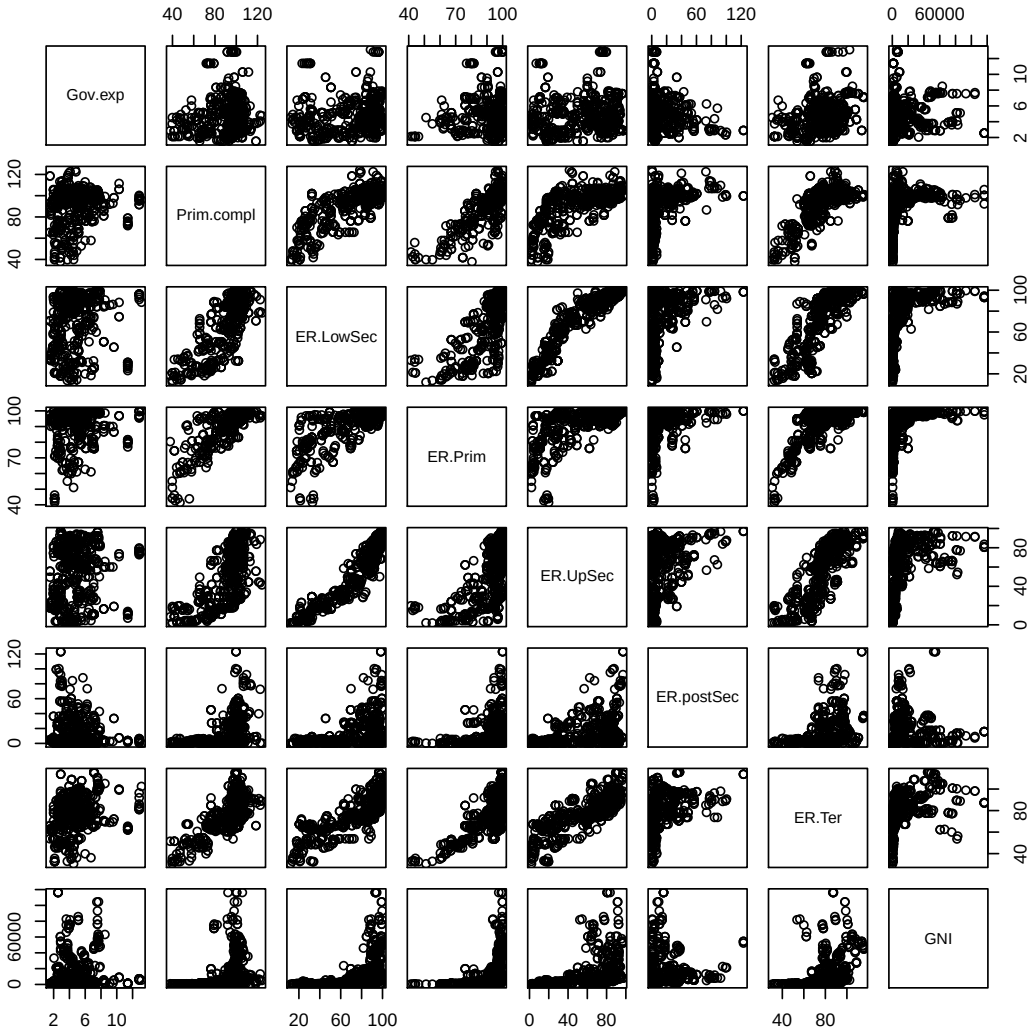


Figura 3.2: A representación das variables enfrontadas dúas a dúas fainos ver que un axuste mediante un hiperplano non dará bos resultados, xa que claramente a variable `GNI` non depende linealmente do resto.

Bibliografía

[1] Abe, S. *Support Vector Machines for Pattern Classification*, 2nd ed., Springer, 2010.

[2] Cristianini, N. and Shawe-Taylor, J. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[3] Dua, D. and Graff, C. *UCI Machine Learning Repository*, 2017. University of California, Irvine, School of Information and Computer Sciences, http://archive.ics.uci.edu/ml

[4] James, G., Witten, D., Hastie, T. and Tibshirani, R. *An Introduction to Statistical Learning with Applications in R*, Springer, 2013.

[5] Kulkarni, A. and Harman, G. *An Elementary Introduction to Statistical Learning Theory*, Wiley Publishing, 2011.

[6] Lin, H. and Lin, C. *A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods*, 2003.

[7] Suykens, J. *A (Short) Introduction to Support Vector Machines and Kernel-based Learning*, ESANN, 2003.

[8] Winston, P. *6.034 Artificial Intelligence*, Fall 2010. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu

[9] Yeh, I., Yang, K. and Ting, T *Knowledge Discovery on RFM Model Using Bernoulli Sequence*, Expert Systems with Applications, 36, 5866–5871.